Universidad Carlos III de Madrid

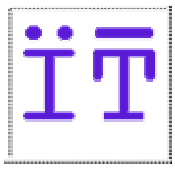
Ingeniería Telecomunicación

Proyecto Fin de Carrera

Integración y pruebas de un sistema de VoIP sobre una red en producción

D. Carlos García García

Octubre de 2001



Universidad Carlos III de Madrid

Ingeniería Telecomunicación

Proyecto Fin de Carrera

Integración y pruebas de un sistema de VoIP sobre una red en producción

Autor: D. Carlos García García

Director: D. José Ignacio Moreno Novella

Octubre de 2001

Departamento de Ingeniería Telemática

PROYECTO FIN DE CARRERA

Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid

Título: Integración y pruebas de un sistema de VoIP sobre una red en producción

Autor: D. Carlos García García

Tutor: D. José Ignacio Moreno Novella

La defensa del presente proyecto fin de carrera se realizó el día 23 de Octubre de 2001 estando el tribunal:

Presidente:

Vocal:

Secretario:

Habiendo obtenido la siguiente calificación:

Presidente **Vocal** **Secretario**

Agradecimientos

Este apartado está dedicado especialmente a mi familia que tanto me ha apoyado en la realización de este proyecto y sin cuyo ánimo diario no habría finalizado el trabajo.

De igual forma quiero agradecer a mi tutor, José Ignacio Moreno, la incalculable ayuda prestada en el desarrollo de todo el proyecto, así como todas las observaciones y correcciones que han permitido enriquecer el presente trabajo. De igual forma le agradezco las oportunidades que me ofreció para trabajar en el Departamento de Ingeniería Telemática.

No puedo olvidar la inestimable ayuda prestada por muchos compañeros de laboratorio que siempre se encuentran dispuestos a echarte una mano en los peores momentos. Quiero destacar la ayuda prestada por mis dos compañeras de laboratorio con quienes comencé a trabajar en el departamento, Mª Carmen y Raquel. Sin olvidar a otros muchos compañeros que conocí con posterioridad como: Jorge, Celia, Cristina, Charly, Ricardo, Manolo, Marcelo, Laura, Juan, Carlos, Oscar, Pablo, etc.

No puedo terminar sin agradecer antes a toda la gente que ha sido la fuente de motivación para la finalización de este proyecto y con quienes he pasado todos estos años de carrera. Esperando no olvidar a nadie: Guillermo, José Emilio, Sara, Pablo, Jose, Chus, Elisa, Patricia, Arancha, Sergio, Gabi, Emilio, Richy, Ricardo, Ismael, Ramón, Adrián, Lucía, Aurora, Edu, Javi, Patricia, Dani,. Y no me quiero olvidar el grupo de Alcorcón: Alfonso, Carlos, Pablo y Pedro.

Índice extendido

Índice abreviado

Resumen

Desde hace tiempo, los responsables de comunicaciones de las empresas comenzaron a tener en cuenta la posibilidad de utilizar su infraestructura de datos para el tráfico de voz interno de la empresa. Sin embargo, la falta de una tecnología adecuada impedía una correcta implantación de esta idea.

En la actualidad, la mejora y abaratamiento de las tecnologías de compresión de voz, así como la estandarización de diferentes protocolos de transmisión de voz por Internet están provocando la implantación de redes VoIP en las empresas.

Tras comprobar que un PC con elementos multimedia puede funcionar como un teléfono a través de Internet, la integración de voz y datos sobre la tradicional red de datos corporativa se situó en el punto de mira de diferentes empresas. Los beneficios que podemos obtener resultan muy interesantes:

- Ahorro de costes de comunicaciones entre las distintas delegaciones de la empresa.
- Integración de servicios y unificación de la estructura.

Es innegable la implantación definitiva del protocolo IP desde los ámbitos empresariales a los domésticos y la aparición de un estándar, el VoIP, no podía hacerse esperar. Para este auge existen otros factores, tales como la aparición de nuevas aplicaciones o la apuesta definitiva por VoIP de fabricantes como Cisco Systems o Nortel-Bay Networks. Por otro lado los operadores de telefonía están ofreciendo o piensan ofrecer en un futuro cercano, servicios IP de calidad a las empresas.

Con este proyecto se pretendía diseñar e implementar una plataforma con soporte VoIP dentro del Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid. Para ello se investigarían todos los elementos de la arquitectura, para posteriormente proceder a la integración de los mismos.

Capítulo 1 - Introducción

El crecimiento y fuerte implantación de las redes IP, tanto en local como en remoto, el desarrollo de técnicas avanzadas de digitalización de voz, mecanismos de control y priorización de tráfico, protocolos de transmisión en tiempo real, así como el estudio de nuevos estándares que permitan la calidad de servicio en redes IP, han creado un entorno donde es posible transmitir telefonía sobre IP.

Si a todo lo anterior, se le suma el fenómeno Internet, junto con el potencial ahorro económico que este tipo de tecnologías puede llevar acarreado, la conclusión es clara: El VoIP (Protocolo de Voz Sobre Internet - Voice Over Internet Protocol) es un tema "caliente" y estratégico para las empresas.

La telefonía sobre IP abre un espacio muy importante dentro del universo que es Internet. Es la posibilidad de estar comunicados a costos más bajos dentro de las empresas y fuera de ellas, es la puerta de entrada de nuevos servicios apenas imaginados y es la forma de combinar una página de presentación de Web con la atención en vivo y en directo desde un call center, entre muchas otras prestaciones. Lentamente, la telefonía sobre IP está ganando terreno... y todos quieren tenerla.

Actualmente, esta tecnología ofrece una calidad de voz de buena a excelente y una fiabilidad de aceptable a muy buena. Algo que, al principio, no siempre sucedía.

Con todo, la interoperatividad entre los productos VoIP sigue siendo un escollo fundamental para la generalización masiva de esta tecnología. El conjunto de estándares englobados en H.323 de la UIT (Unión Internacional de Telecomunicaciones), el primero obtenido para asegurar la interoperatividad en voz sobre IP, se ha mostrado difícil y complejo de implementar. Como resultado, han aparecido otras normas más manejables, sin que hasta ahora se sepa claramente cuál será la más implementada.

No obstante, poco a poco comienza a verse un cierto consenso dentro de la comunidad de fabricantes sobre el futuro de los distintos estándares. Según la evolución del mercado que hemos visto en los últimos años, se puede afirmar que coexistirán diferentes normas, entre ellas H.323 de UIT, SIP (Session Initiation Protocol) y MGCP (Media Gateway Control Protocol) del IETF (Internet Engineering Task Force), y H.248/Megaco, también de la UIT. Así, pues, no es de esperar, al menos a corto plazo, que un sólo estándar se imponga como el claramente dominante.

1.1. Objetivos

El objetivo del presente proyecto es la creación y mantenimiento de una plataforma piloto de voz sobre IP que permita la realización de pruebas y medidas de calidad, así como la posibilidad de incorporar nuevos elementos y desarrollar servicios que doten a la plataforma de mayor interoperabilidad. De esta manera la plataforma quedaría disponible para realizar llamadas dentro del Departamento de Ingeniería Telemática, así como para disponer de una plataforma estable para la realización de pruebas y desarrollo de servicios avanzados sobre la misma.

En este ámbito, y como objetivo del proyecto se propuso la incorporación de soporte para la versión 6 del protocolo IP. De esta manera se pretendía conseguir una plataforma de telefonía con soporte IPv6. Este objetivo podría alcanzarse mediante la adaptación de desarrollos SIP existentes.

Para acometer los diferentes propósitos que acabamos de comentar, dividimos el proyecto en diferentes tareas que permitiesen afrontar los diferentes objetivos. En primer lugar deberíamos realizar un estudio completo sobre el estado de la tecnología de voz sobre IP, lo cual nos permitiese identificar los diferentes componentes que formarían la plataforma. Posteriormente se debería realizar una búsqueda adecuada para localizar el hardware necesario, así como proyectos que desarrollasen las aplicaciones software necesarias. Una vez realizadas estas etapas procederíamos a la integración de todos los componentes para el montaje de la plataforma. Y finalmente podríamos realizar diferentes baterías de pruebas para medir la calidad de comunicación alcanzada.

En segundo lugar para lograr la integración de la plataforma en un entorno SIP sobre IPv6, realizaríamos en primer lugar un estudio sobre el estado del arte. Posteriormente se procedería a realizar un estudio del software propuesto para la migración IPv4-IPv6, y tras comprobar la viabilidad de este cambio se procedería a realizar los cambios oportunos y probar el funcionamiento de las aplicaciones en el nuevo escenario.

Resulta muy interesante destacar la fuerte realimentación existente entre las diferentes tareas, de manera que la consecución completa o parcial de alguna de estos puede determinar la variación de algunos de los puntos que definen el resto de objetivos descritos.

1.2. Estructura, Fases y Medios Materiales

Definiremos a continuación la estructura del proyecto determinada para la consecución de los objetivos anteriormente descritos.

ESTRUCTURA
Implementación de una plataforma piloto de voz sobre IP - Localización de los componentes necesarios - Integración de los mismos en base a unos requisitos establecidos **Realización de pruebas de calidad de servicio sobre la plataforma** - Medida de los niveles de calidad en función de los distintos parámetros configurables que pueden modificarse en la plataforma. - Estudio de la viabilidad de la comunicación entre diferentes emplazamientos en función del acceso a red disponible, así como la distancia entre puntos de acceso. **Incorporación a la plataforma de soporte sobre una red IPv6** - Estudio de la viabilidad de migración de las aplicaciones existentes en la plataforma del protocolo de comunicaciones IPv4 a su sucesor IPv6. - Implementación de estos cambios sobre un cliente de voz sobre IP, y comprobación de su correcto funcionamiento.

Tabla I.1. Estructura designada para la realización del proyecto

Una vez descrita la estructura del proyecto podemos apreciar las diferentes fases en que se ha descompuesto. Cabe destacar que la siguiente distribución se ha realizado a posteriori ya que resulta imposible determinar desde un principio cual será la evolución del proyecto y en que nuevas tareas puede desembocar.

Fases del proyecto
Fase I - Análisis del estado del arte sobre tecnología VoIP: posibles escenarios para un estudio de las características de comunicación, componentes que integran una plataforma VoIP, otros proyectos que desarrollen este tipo de aplicaciones. **Fase II** - Análisis, diseño e implementación de la plataforma piloto VoIP. - Medida de las principales características del escenario generado. **Fase III** - Estudio sobre las características de migración de aplicaciones entre IPv4 e IPv6. - Migración de una aplicación en concreto dentro de nuestra plataforma.

Tabla I.2. Fases que componen el proyecto

Para finalizar este apartado vamos a describir los medios materiales que han sido necesarios para llevar a cabo este proyecto.

Medios Materiales

- Infraestructura de comunicaciones y acceso a Internet del Departamento de Ingeniería Telemática
 	
- Tres ordenadores personales que actúen como:
  o Clientes H.323 (Windows, Linux)
  o Gatekeeper trabajando en entorno Linux
  o Entorno de desarrollo
 	
- Dos router Nucleox+ de TELDAT configurados como Gateways de la plataforma que incluyen:
  o Una línea telefónica conectada a la centralita de la universidad
 	
- Bibliografía disponible en la biblioteca de la Universidad Carlos III.

Tabla I.3. Medios Materiales utilizados durante la realización del proyecto

1.3. Organización de la Memoria

En este apartado indicaremos brevemente el contenido de los diferentes capítulos que forman la memoria de este proyecto.

En el presente capítulo se indica el objetivo del proyecto, que será el que determine la evolución del mismo, además de incluir el estado del arte de la tecnología empleada. También se incluye la estructura del proyecto así como las fases seguidas en su evolución.

En el capítulo segundo, “Voz sobre IP” se realiza un estudio sobre las diferentes tecnologías presentes en la actualidad que permiten el soporte de “Voz sobre IP”. Igualmente se analizan todos los factores a tener en cuenta en el desarrollo de una plataforma de estas características. Dentro de cada tecnología estudiada se presentan los diferentes componentes que la forman orientándolo a la posterior implantación sobre una plataforma experimental.

El capítulo tercero se titula “Desarrollo de una plataforma VoIP” y en el se expone todo el desarrollo llevado a cabo para la implementación de la plataforma piloto de pruebas para voz sobre IP. Para ello se estudian los diferentes componentes que finalmente pasaron a formar parte de la plataforma. En el final de este capítulo se incluye el estudio de calidad realizado sobre la plataforma, así como otro estudio sobre la viabilidad de la comunicación VoIP entre diferentes emplazamientos.

A través del capítulo cuarto, “Integración de la plataforma en un entorno IPv6” se pretende explicar la que podría considerarse segunda fase por la que paso el proyecto. En ella se explican las diferentes fases empleadas para lograr la migración de aplicaciones pertenecientes a la plataforma desde IPv4 hasta IPv6. Previamente se realiza un estudio teórico sobre el estado actual del transito entre estas dos versiones del conocido protocolo de comunicaciones.

En el capítulo quinto, “Conclusiones y trabajos futuros” se discuten los logros alcanzados mediante la realización de este proyecto. De igual forma se intentan fijar futuras líneas de trabajo sobre los desarrollos realizados.

Finalmente en los apéndices se incluyen el presupuesto del proyecto, donde se desglosan las diferentes tareas realizadas y se identifican los costes en que ha incurrido el proyecto; así como diferentes manuales de instalación y configuración relativos a componentes de la plataforma VoIP.

Capítulo 2 – Voz sobre IP

El desarrollo de las telecomunicaciones y en particular de Internet ha hecho que tecnologías como la telefonía IP (Internet Protocol) comiencen a ser una realidad tanto en el mundo de los negocios como del ocio. Los problemas generados por la heterogeneidad del gran número de redes de telecomunicaciones existentes están motivando el estudio de mecanismos que favorezcan la homogeneización de los medios de transporte de voz y datos.

2.1.- Introducción a VoIP (Voz sobre IP)

La convergencia de las redes de telecomunicaciones actuales supone encontrar la tecnología que permita hacer convivir en la misma línea la voz y los datos. Esto obliga a establecer un modelo o sistema que permita "empaquetar" la voz para que pueda ser transmitida junto con los datos. Teniendo en cuenta que Internet es la "red de redes", desarrollar una tecnología de ámbito mundial nos dirige claramente al protocolo IP (Internet Protocol) y a encontrar el método que nos permita transmitir voz a la vez que datos sobre ese protocolo. El problema tiene una "sencilla" solución: VoIP (Voice Over Internet Protocol).

Mediante la realización del presente trabajo se pretende analizar si el camino hacia la convergencia está bien diseñado, cuáles son las ventajas e inconvenientes del mismo y exponer las conclusiones de un caso práctico en el cual se ha utilizado la última tecnología existente en la actualidad. Si realmente la convergencia es posible, la interpretación del trabajo y la adaptación a las distintas situaciones actuales puede ser un primer paso que se habrá dado para alcanzar cuanto antes el futuro más inmediato.

Algo tan sencillo en principio no lo es en la realidad y para comprobarlo sólo hay que repasar la evolución de los distintos desarrollos comerciales, de los distintos estándares y las distintas nomenclaturas y acrónimos que utilizan todos los expertos en la materia .

Este trabajo basado en una experiencia real sobre VoIP intenta analizar la situación actual con respecto a estudios e investigaciones previas para así poder plantear opciones de futuro y su aplicación a la realidad empresarial.

Aunque son conocidas distintas investigaciones en algoritmos avanzados de digitalización de voz desde 1970 y distintas experiencias de transmisión de voz sobre redes locales (LAN) en los años 80, es en Febrero de 1995 cuando la empresa VocalTec da el pistoletazo de salida mostrando a través de su producto Internet Phone las posibilidades reales de establecimiento de llamadas telefónicas de Pc a Pc. Se utilizaba

entonces un paquete de software instalado en el Pc y como medio de transmisión Internet. Nacía así el término hoy acuñado como Telefonía IP.

La evolución en el tiempo ya era imparable y es en 1996 cuando se dan las primeras experiencias de establecimiento de llamadas de Teléfono a Pc y de Teléfono a Teléfono. A partir de 1997 empiezan a aparecer nuevos dispositivos y métodos que nos llevan hoy en día a mantener el término XoIP ('X' over Internet Protocol) como la verdadera opción de futuro o si se prefiere como la puerta hacia la convergencia de las redes. En este acrónimo X significa cualquier contenido susceptible de ser transmitido por una red (D = data, V = voz, F = fax, M = multimedia, etc).

Este laberinto de tecnologías, de intereses comerciales y de opciones de futuro lleva como toda "revolución" a la confusión y desgaste del público en general. La consecuencia inmediata son las habituales FAQs (Frequently Asked Questions): ¿por qué IP?, diferencia entre Telefonía IP y Voz sobre IP, ¿es VoIP lo mismo que VoFR (Voz sobre Frame Relay)?, ¿qué significa realmente XoIP?, etc.

Es preciso, por tanto, definir de una forma simple y clara la situación actual para que a partir de este momento se puedan identificar claramente tanto los términos como los elementos que de alguna u otra forma intervienen en los distintos niveles del desarrollo de la convergencia de redes. Términos que posiblemente identifican el camino hacia los servicios de VoIP:

- Telefonía: servicios de telecomunicación prestados sobre la Red Telefónica Conmutada (RTC) ya sea Red Telefónica Básica (RTB) o Red Digital de Servicios Integrados (RDSI), a excepción de comunicación de datos.
- Voz en Internet: servicios de telefonía prestados sobre la red pública global formada por la interconexión de redes de conmutación de paquetes basadas en IP.
- Voz sobre IP (VoIP): servicios de telefonía prestados sobre redes IP "privadas" sin interconexión a la RTC
- Telefonía IP: servicios de telefonía prestados sobre Redes IP "privadas" en interconexión con la RTC.

- Voz sobre Frame Relay (VoFR): servicios de telefonía prestados sobre redes soportadas por circuitos Frame Relay, orientados a la transmisión de datos.
- Voz sobre ATM (Asyncronus Transfer Mode) (VoATM): servicios de telefonía prestados sobre redes ATM donde existe posibilidad de ofrecer una calidad de servicio (Qos).
- Multimedia sobre IP (MoIP): servicios multimedia (vídeo, audio, imagen, etc) prestados sobre redes IP
- Fax sobre IP (FoIP): servicios de transmisión de fax prestados sobre redes IP.
- XoIP: en términos globales "todo sobre IP". Se trata de sustituir X por aquella letra que identifique cualquier servicio sobre redes IP (F = fax, M = multimedia, V = voz, D = data, etc).

Como conclusión se puede deducir que si el futuro es IP (debido sobre todo a su ámbito de cobertura actual, su aceptación por parte del usuario y la próxima aparición del protocolo IPv6) y que si X es la integración global de todos los servicios actuales y de futuro, XoIP es el verdadero camino que puede abrir las puertas hacia la Convergencia de Redes. Esta Convergencia supone la unificación sobre una misma estructura de la transmisión de voz y datos. La convergencia supondrá en términos económicos una auténtica "revolución" que afectará desde el entorno empresarial hasta el entorno doméstico. La reducción de costes en todos los ámbitos se puede considerar como inaudita.

A continuación se compara la telefonía IP con la tradicional y se describen las ventajas e inconvenientes de los servicios IP.

2.1.1- Telefonía IP vs. Telefonía tradicional

Aunque la telefonía IP aprovecha la infraestructura de telecomunicaciones ya existente necesita nuevos elementos.

En la figura II.1 se puede apreciar la realidad actual, un entorno en donde conviven de forma paralela las redes de una determinada organización. Por un lado existe un circuito de datos y de forma paralela se aprecia un circuito de voz.

Lo que proponemos es la incorporación de unos elementos denominados VoIP GW (Gateway o Pasarela para Voz sobre IP) mediante los cuales se consigue la unificación de ambas redes y por tanto se logra la Convergencia.

La telefonía IP, necesita un elemento que se encargue de transformar las ondas de voz en datos digitales y que además los divida en paquetes susceptibles de ser transmitidos haciendo uso del protocolo IP. Este elemento es conocido como Procesador de Señal Digital (DSP), el cual está ya disponible y utilizan las Teléfonos IP o las propias Gateways o Pasarelas encargadas de transmitir los paquetes IP una vez paquetizada la voz. Cuando los paquetes alcanzan el Gateway de destino se produce el mismo proceso a través del DSP pero a la inversa con lo cual el receptor podrá recibir la señal analógica correspondiente a la voz del emisor.

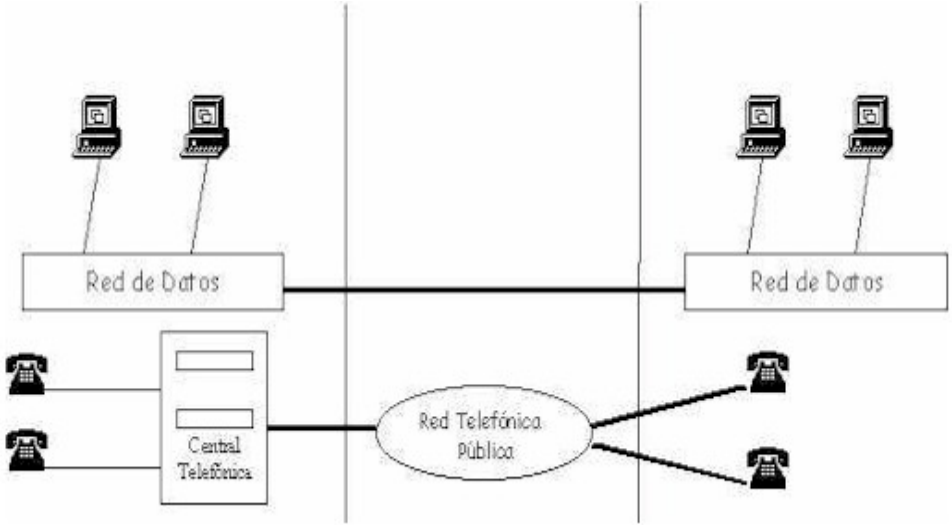


Figura II.1: Infraestructura de red actual correspondiente

La transmisión de paquetes de voz según la forma expuesta, es similar a la transmisión de un correo electrónico desde el origen hasta el destino. El problema es que en las transmisiones IP no está garantizado el éxito, por lo cual si el correo no es legible o se "pierde" algún paquete, es necesario solicitar la retransmisión del mismo y su recuperación es factible. Pero en el caso de la transmisión de voz esto no es así, ya que la necesidad de recibir los paquetes en un determinado orden, la necesidad de asegurar que no haya pérdidas y de conseguir una tasa de transmisión mínima hacen prácticamente necesaria la implantación de sistemas de Calidad de Servicio (QoS: Quality of Services). Estos sistemas suponen hoy en día el gran reto de la industria ya que garantizar "Quality of Service Over IP" supondrá la inmediata implantación de los sistemas de transmisión de voz.

A modo de resumen el verdadero problema hoy en día es que la Telefonía Conmutada establece circuitos virtuales dedicados entre el origen y el destino y ahí la calidad es innegable y segura. Por el contrario la transmisión de voz sobre IP comparte el circuito y el ancho de banda con los datos y los paquetes pueden atravesar multitud de nodos antes de llegar a su destino lo que supone lógicas deficiencias en la transmisión de paquetes de voz.

A continuación se plantean otras cuestiones referentes a esta tecnología y que tienen que ser obligatoriamente consideradas a la hora de llevar a cabo una posible implantación real de un sistema de telefonía IP para uso comercial o profesional:

2.1.2.- Calidad de transmisión

Hasta hace muy poco tiempo el ancho de banda necesario para la transmisión de voz y vídeo en tiempo real era considerablemente elevado, lo que hacia imposible este tipo de comunicaciones sobre redes de datos que no garantizaran una calidad de servicio, como por ejemplo Internet o redes basadas en protocolo IP.

Actualmente la voz que recibe un gateway es digitalizada y comprimida según distintos algoritmos (GSM, G.723.1, G.711, G.729) los cuales se caracterizan por conseguir mayores ratios de compresión en detrimento del tiempo de latencia (tiempo necesario para descomprimir la voz para que pueda ser entendida de nuevo). Algunos de estos algoritmos consiguen comprimir los paquetes de voz en 8 Kbps aproximadamente. El protocolo IP añade al paquete de voz digitalizado y comprimido una serie de cabeceras para su correcto transporte a través de la red, lo que hace que el ancho de banda necesario se incremente hasta unos 16 Kbps.

Hay que considerar así mismo el parámetro denominado "supresión de silencio". Con este parámetro activado, se consigue que la transmisión de paquetes (uso de ancho de banda) se reduzca a las situaciones en que los agentes están hablando. El resto del tiempo (cuando no existe voz a transmitir) se libera el ancho de banda. Considerando este aspecto, se puede afirmar que el tamaño medio de un paquete de voz durante una conversación es de 8 Kbps.

Con todo lo anterior se puede afirmar que con un canal B de cualquier línea RDSI (Red Digital de Servicios Integrados: 2 canales B y 1 canal D), cuyo ancho de banda es de 64 Kbps se puede realizar una comunicación de 8 llamadas simultáneas. Esta situación suele coincidir con las dimensiones de cualquier centralita de una PYME (Pequeña y Mediana Empresa). Esto viene a demostrar que las necesidades de ancho de banda para este tipo de aplicaciones está al alcance de prácticamente cualquier empresa.

Referente a la calidad de la transmisión de la voz, todos los fabricantes e investigaciones hacen referencia a tres factores determinantes:

- Codificadores de Voz: influyen en la digitalización de la voz en paquetes de datos que contienen voz y que serán transmitidos por la red IP, también influyen por el retardo necesario para la descompresión de esos paquetes voz, lo que imputa un retardo añadido a la comunicación.
- Cancelación de Eco: requerimiento necesario para una comunicación a través de Telefonía IP, que elimina de forma automática y en tiempo real posibles ecos, ya que si no lo hiciera haría inteligible la comunicación.
- Latencia: tiempo necesario para que la voz viaje de un extremo al otro, incluyen los tiempos necesarios para la compresión, transmisión y descompresión. Este tiempo tiende a minimizarse pero jamás podrá ser suprimido. Actualmente los tiempos que se están obteniendo de latencia giran alrededor de 120 ms.

2.1.3.- Estándares

Actualmente existen estándares que regulan este tipo de comunicaciones, estándares que provienen de organismos internacionales de estandarización como el ITU (International Telecommunication Union) que ha establecido unas normas para la interconexión de los distintos elementos que intervienen en una comunicación sobre Telefonía IP.

El estándar que regula este tipo de comunicaciones es el H.323 de la ITU (ITU Standards, 1998). Esta norma realmente es una serie de normas para la transmisión de datos multimedia (audio, vídeo y datos) sobre redes que no garantizan una calidad de servicio (redes IP).

Las funciones cubiertas por el H.323 son acerca del control de llamadas, uso de codificadores de voz y normas de otros organismos que especifican la transmisión en tiempo real de los paquetes de voz.

El protocolo H.323 ha sido adoptado prácticamente por todas las empresas líderes en este sector como Netscape, Microsoft, Intel, Vocaltec. La adopción de este estándar permite la interconexión de equipos y software de cualquier fabricante que lo haya adoptado.

Por tanto es lógico deducir que en la actualidad cualquier empresa que quiera trabajar en servicios de VoIP debe adoptar este estándar en todos sus desarrollos. De esta manera se garantizará una perfecta integración con plataformas hardware y software de distintos fabricantes cuyos productos sigan la misma norma.

2.1.4.- Regulaciones Gubernamentales

Los servicios de Voz sobre IP inquietan al gobierno español, debido al rápido crecimiento que está teniendo Internet y la posibilidad de ofrecer cantidad de servicios de valor añadido sobre esta red, como puede ser la telefonía IP.

A este respecto podemos considerar la posición de la Unión Europea, que ha establecido diversos criterios que la telefonía IP debería tener antes de que pueda estar sometida a regulación:

- Que las comunicaciones sean objetos de ofertas comerciales.
- Que las comunicaciones sean suministradas para el publico en general.
- Que las comunicaciones tengan como origen y destino puntos de la red publica telefónica conmutada.
- Que la comunicación implique transporte y conmutación en tiempo real.

La Unión Europea considera hoy en día que estos cuatro criterios no se cumplen, y por ello la Unión Europea ha decidido no regular la telefonía IP. Esto demuestra que la Unión Europea ha subestimado el potencial de desarrollo de esta tecnología ya que estas características ya son posibles con la tecnología existente hoy en día.

En lo que respecta al gobierno español, su postura está en la misma línea que la Unión Europea, subestima las posibilidades de desarrollo de esta tecnología, pero asume la decisión de este organismo y tampoco regulará este tipo de comunicaciones.

Es necesario indicar que esta situación legal evidentemente es susceptible de cambio en cualquier momento a criterio de los organismos reguladores competentes en esta materia.

2.1.5.- Aplicaciones

Con todo lo anteriormente descrito, se pueden poner en marcha una serie de aplicaciones que son de gran demanda que producen de forma inmediata un ahorro de costes muy significativo.

Centros de llamadas (Call centers):

Los centros de llamadas pueden usar la Telefonía IP, mejorando la calidad de la información intercambiada en cada sesión. Por ejemplo un usuario podría navegar por información on-line, antes de realizar la consulta a un operador. Una vez en comunicación con el operador, se podría trabajar con un documento compartido a través de la pantalla. De esta forma se consigue sistemas de una gran calidad en el servicio a ofrecer, además de reducir de forma considerable el coste de líneas telefónicas y de Distribuidores Automáticos de Llamadas (ACD).

Redes Privadas virtuales de Voz:

Esta aplicación consiste en la interconexión de las centralitas telefónicas a través de la red IP corporativa, de manera que se puede realizar una llamada desde una extensión de la oficina A otra extensión de la oficina B a través de la red de datos de la empresa, produciéndose esta llamada de forma gratuita ya que se aprovecha la infraestructura de datos ya existente. Un ejemplo claro de este servicio serían los bancos y su red de oficinas.

Centros de llamadas por el WEB:

Si una compañía tiene su información disponible en un Web en Internet, los usuarios que visitan este Web podrían no solo visualizar la información que esta compañía les ofrece, sino que podría establecer una comunicación con una persona del departamento de ventas sin necesidad de cortar la conexión. De esta manera el operador de ventas cuando atienda la llamada tendrá en su pantalla la misma información que esta viendo el usuario. Esta aplicación tiene las siguientes ventajas:

- Al ser la llamada a través de Internet, para el usuario no tiene coste adicional, aprovecha la llamada telefónica que tenía establecida para la comunicación de datos, para mantener también la comunicación de voz, esto permite tener a la empresa un servicio similar al de las líneas 900.
- El usuario puede mantenerse on-line mientras habla con un operador de ventas.
- El cliente trata con operadores humanos, que le podrán asesorar, esta característica mejorará sin lugar a duda el resultado de un sistema de comercio electrónico.
- El operador puede cerrar la venta de manera más fácil ya que el usuario es bastante reacio a dar los datos de su tarjeta de crédito en una pagina Web por temas de seguridad que todos conocen, sin embargo no tendrá ningún inconveniente de dar esos datos verbalmente al operador de ventas, teniendo el usuario plena garantía de que sus datos están a salvo.

Aplicaciones de FAX:

Al igual que se hace con la voz, cabe la posibilidad de realizar transmisiones de FAX sobre redes de Telefonía IP, consiguiendo de esta manera reducir de forma significativa los costes de una empresa en transmisión de fax. En este caso no es necesario para el usuario que recibe el fax de disponer de equipos especiales ya que los faxes se seguirán recibiendo a través de una máquina de fax convencional. Una aplicación típica en este tema es el envío masivo de fax, ya que el usuario sólo enviará una copia del fax que desea enviar, así como la lista de números telefónicos de destino y el sistema se encargará de realizar todos los envíos enrutando los faxes al punto desde donde la llamada de destino es más económica.

Multiconferencia:

La telefonía IP permite la conexión de 3 o más usuarios simultáneamente compartiendo las conversaciones de voz o incluso documentos sobre el que todos los miembros de la multiconferencia pueden participar en la revisión, esto resulta de gran utilidad para empresas que realicen reuniones virtuales, con los consiguientes ahorro de gastos que supone el desplazamiento de personas.

2.1.6.- Ventajas e Inconvenientes de los Servicios IP

En esta sección se analizan por separado tanto las ventajas como los inconvenientes del uso de los servicios IP en los ámbitos más comunes. Así mismo se analizan los aspectos más relevantes que impiden una rápida implantación de estos servicios:

Ventajas:

Los servicios de VoIP presentan una multitud de ventajas en todos los aspectos. Su enumeración y explicación debe de realizarse de forma sencilla y transparente al objeto de hacer llegar a los posibles usuarios la bondad de su implantación en un futuro no muy lejano. Hay que evitar la confusión y prematuro rechazo ante algo que se plantea como la solución universal y que no se termina de entender. En esta línea destacan tres grandes bloques:

- Entorno empresarial:

1. Amplia reducción en los costes de la factura telefónica. Los costes de todo tipo de llamadas se equipararán al de una llamada local de forma que la reducción en los costes del tráfico de voz será a todas luces muy importante
2. Nuevas posibilidades de marketing directo y potenciación del servicio de atención al cliente. Podrán implantar la filosofía "Push 2 Talk" que consiste en un icono situado en una página Web a través del cual un navegante podrá dialogar con personal especializado de la compañía mientras continúa navegando por la red.
3. Potenciación del teletrabajo y de los teletrabajadores. Con una única conexión se podrá acceder a aplicaciones corporativas, al correo vocal, atender llamadas o buscar información sobre nuevos proyectos.

- Usuarios Finales:

1. En este momento el usuario final que ocupe su línea de teléfono doméstica para transmisión de datos no puede recibir comunicaciones de voz al estar la línea ocupada. Los nuevos servicios de VoIP no sólo le permitirán atender llamadas de forma simultánea sino que además podrá conocer quien le llama y de esa forma admitir y rechazar llamadas e incluso desviarlas.

- Proveedores de Servicios:

1. XoIP será su nuevo argumento comercial. X supone poder ofrecer voz, datos, fax o cualquier servicio susceptible de ser transmitido por una red IP. El ejemplo más claro es la nueva vertiente estadounidense denominada Internet Telphony Service Providers (ITSPs) quienes ya ofrecen todo tipo de servicios a través de redes IP.

Inconvenientes

Si todo está tan claro, si ya existe tecnología, si los estándares están validados por organismos internacionales (caso del H.323 definido por la ITU), si la ley en principio no presenta inconvenientes y si además las consultoras internacionales presentan esta solución como la verdadera alternativa de negocio en el año 2005, la lógica hace pensar que la implantación de XoIP se realizará de forma inmediata. Pero el verdadero caballo de batalla se resume con tres letras "QoS".

Quality of Service: garantizar calidad de servicio en base a retardos y ancho de banda disponible en una red IP no es realmente posible sobre una red IP. Una vez digitalizada la voz y paquetizada, se envía al canal de transmisión y aquí no existen soluciones que nos garanticen o permitan establecer anchos de banda, orden de paquetes y retrasos asumibles en su transmisión. Las posibles soluciones pasan por diferenciar los paquetes de voz de los paquetes de datos, priorizar la transmisión de los paquetes de voz y hacer que los retrasos añadidos a la transmisión de los paquetes no superen en ningún caso los 150 milisegundos (recomendación de la ITU).

Las líneas de trabajo actuales y las soluciones hasta el momento desarrolladas, se basan en:

- Anchos de Banda:

  En la tabla II.1 se muestra la relación existente entre los distintos algoritmos de compresión de voz utilizados y el ancho de banda requerido por los mismos:

VoCodecs	Ancho de Banda (BW)
G.711 PCM	64 kbps
G.726 ADPCM	16, 24, 32, 40 kbps
G.727 E-ADPCM	16, 24, 32, 40 kbps
G.729 CS-ACELP	8 kbps
G.728 LD-CELP	16 kbps
G.723.1 CELP	6.3 / 5.3 kbps

Tabla II.1: Ancho de Banda requerido por los Codecs actuales

- Retardo:

  Una vez establecidos los retardos de procesado, retardos de tránsito y el retardo de procesado la conversación se considera aceptable por debajo de los 150 ms.

- Eco:

  El eco es debido a una reflexión, habitualmente se debe a un desajuste de impedancias, si bien en el caso de VoIP debemos incluir el eco debido a la utilización de micrófono y altavoces en la comunicación.

- Obtener QoS:

  Las líneas de trabajo actuales de cara a conseguir Calidad de Servicio en una Transmisión IP, están basadas en:

  a.-Supresión de silencios y VAD (voice activity detection): establecer diferencia entre habla y silencio, no transmitir paquetes de silencio y generación de silencios al otro extremo.

  b.-Reserva de Ancho de Banda: implantación del estándar RSVP (Protocolo de Reserva de Recursos) de la IETF (Internet Engineering Task Force). RSVP incorpora reserva de ancho de banda y retardo además de establecer una lista de acceso dinámica de extremo a extremo. Sus principales deficiencias se establecen en su defectuoso crecimiento (solución válida en redes pequeñas) y en su deficiente autorización y autentificación. Además hay que tener en cuenta que las actuales infraestructuras no la tienen en cuenta. Para solucionar los problemas que conlleva RSVP aparecen técnicas como Diffserv, basadas en distinguir diferentes tipos de flujos y otorgarles diferentes prioridades.

  c.-Priorizar: existen diferentes tendencias tales como:

    1.- CQ (Custom Queuing): asignación de un porcentaje del ancho de banda disponible.

    2.- PQ (Priority Queuing): establecer prioridad en las colas.

    3.- WFQ (Weight Fair Queuing): asignar prioridad al tráfico de menos carga.

    4.- DiffServ: definido en borrador por la IETF, evita tablas en routers intermedios y establece decisiones de rutas por paquete.

  e.-Control de Congestión: uso del protocolo RED (Random Early Discard), técnica que fuerza descartes aleatorios.

2.2.- H.323

2.2.1.- Introducción a H.323

El H.323 es una familia de estándares definidos por el ITU para las comunicaciones multimedia sobre redes LAN. Está definido específicamente para tecnologías LAN que no garantizan una calidad de servicio (QoS). Algunos ejemplos son TCP/IP e IPX sobre Ethernet, Fast Ethernet o Token Ring. La tecnología de red más común en la que se están implementando H.323 es IP (Internet Protocol).

Este estándar define un amplio conjunto de características y funciones. Algunas son necesarias y otras opcionales. El H.323 define mucho más que los terminales. El estándar define los siguientes componente más relevantes: Terminal, GateWay, Gatekeeper, Unidad de Control Multipunto (MCU).

El H.323 utiliza los mismos algoritmos de compresión para el vídeo y el audio que la norma H.320, aunque introduce algunos nuevos. Se utiliza T.120 para la colaboración de datos.

El H.323 es la primera especificación completa bajo la cual, los productos desarrollados se pueden usar con el protocolo de transmisión más ampliamente difundido (IP). Existe tanto interés y expectación entorno al H.323 porque aparece en el momento más adecuado. Los administradores de redes tienen amplias redes ya instaladas y se sienten confortables con las aplicaciones basadas en IP, tales como el acceso a la web. Además, los ordenadores personales son cada vez más potentes y , por lo tanto, capaces de manejar datos en tiempo real tales como voz y vídeo.

Varias compañías consultoras independientes predicen una rápida adopción del H.323. El gráfico II.2 explica por sí mismo esta tendencia.

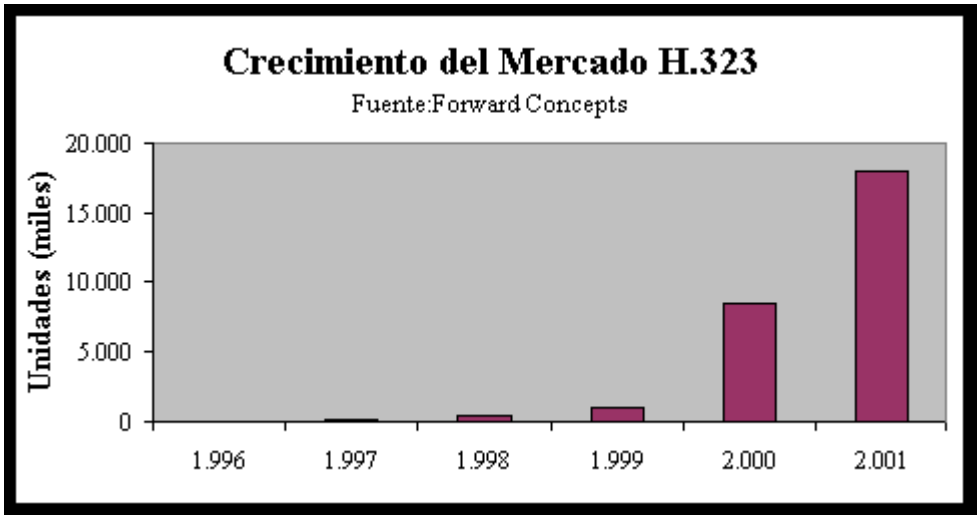


Figura II-2 – Crecimiento del mercado H.323

La existencia del H.323 es una "apuesta a caballo ganador" para los usuarios. Una de sus más importantes ventajas es la interoperabilidad de los equipos. Dentro de una única red, los sistemas H.323 de diferentes fabricantes serán intercambiables. Un gateway de un determinado fabricante puede coexistir y trabajar junto con terminales de diferentes fabricantes. La conectividad fuera de la propia red también (con clientes, proveedores, etc.) se simplifica notablemente. La existencia de un estándar impulsa la competencia y produce un ajuste de precios.

Este optimismo general del mercado debe ser contemplado cuidadosamente para no caer en algunas falsedades difundidas en torno a la tecnología de vídeo sobre IP. Es posible que eventualmente todos los ordenadores con un puerto LAN lleguen a tener capacidades de vídeo. Sin embargo, el nivel de prestaciones de estos equipos estará en muchos casos limitadas, aunque mejoren conforme lo hace la tecnología de los PC´s y los procesadores digitales de señal. Los fabricantes, por su parte, introducirán con el tiempo diversas soluciones de valor añadido. La variedad de terminales H.323 combinada con adaptadores, gateways y otros productos de infraestructuras nos puede proporcionar una conectividad universal dentro y fuera del ámbito de una misma empresa.

2.2.2.- Componentes de la arquitectura H.323

Un *terminal* H.323 es un extremo de la red que proporciona comunicaciones bidireccionales en tiempo real con otro terminal H.323, gateway o unidad de control multipunto (MCU). Esta comunicación consta de señales de control, indicaciones, audio, imagen en color en movimiento y /o datos entre los dos terminales. Conforme a la especificación, un terminal H.323 puede proporcionar sólo voz, voz y datos, voz y vídeo, o voz, datos y vídeo. La estructura de un terminal H.323 se muestra en la figura II.3.

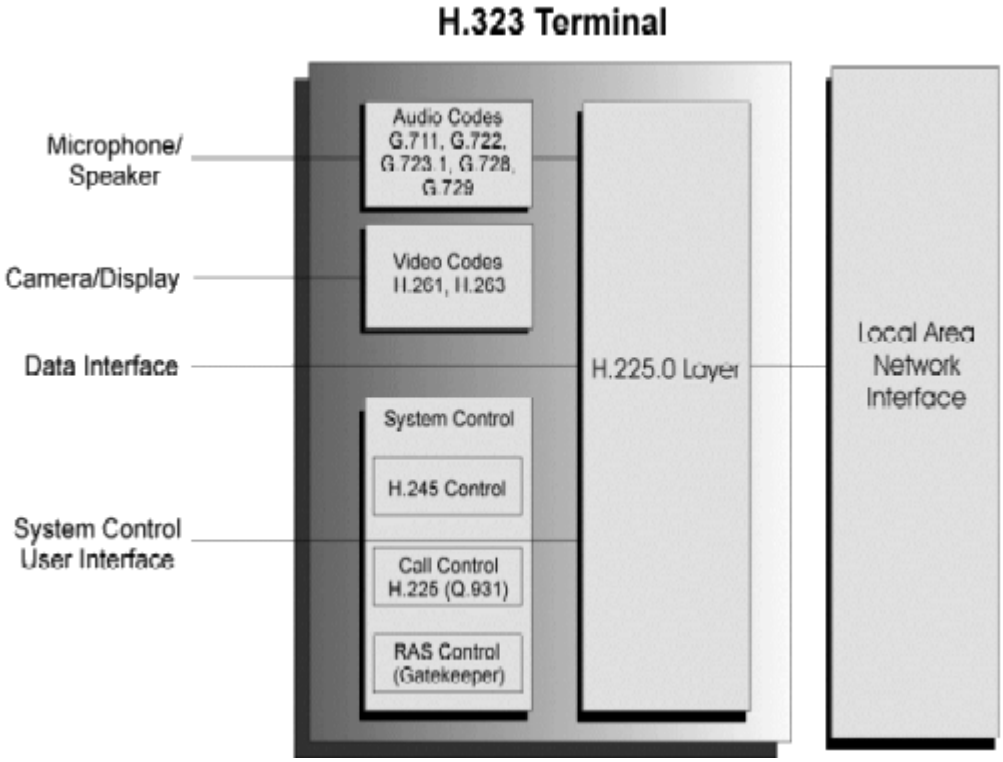


Figura II.3: Estructura de un Terminal H.323

Todos los terminales deben tener una unidad de control del sistema, una capa H.225.0, una interfaz de red y un codificador de audio. El codificador de vídeo y las aplicaciones de datos son opcionales.

El gatekeeper (GK) es una entidad que proporciona la traducción de direcciones y el control de acceso a la red de los terminales H.323, gateways y MCUs. El GK puede también ofrecer otros servicios a los terminales, gateways y MCUs, tales como gestión del ancho de banda y localización de los gateways o pasarelas.

Un gateway H.323 (GW) es un extremo que proporciona comunicaciones bidireccionales en tiempo real entre terminales H.323 en la red IP y otros terminales o gateways en una red conmutada. En general, el propósito del gateway es reflejar transparentemente las características de un extremo en la red IP a otro en una red conmutada y viceversa. En otras palabras, nos servirá de pasarela entre el entorno de vídeo sobre IP H.323 y el entorno vídeo sobre RDSI H.320.

Una unidad de control multipunto H.323 (MCU) es un extremo que proporciona la capacidad para que tres o más terminales y gateways participen en una conferencia multipunto. La MCU opera generalmente como una MCU H.320, aunque no es obligatorio un procesador de audio. Una MCU se forma de dos partes: un controlador multipunto (MC) que es obligatorio y un procesador multipunto (MP) opcional. En el caso más simple, una MCU puede estar formada por un MC únicamente.

Un controlador multipunto (MC) es una entidad H.323 que proporciona las capacidades de negociación entre todos los terminales para conseguir la comunicación. Puede controlar así mismo recursos de la conferencia tales como el vídeo multicast. El MC no realiza mezcla ni conmutación de audio, vídeo o datos.

Un procesador multipunto (MP) es la entidad H.323 cuyo hardware y software especializado mezclan, conmutan y procesan el audio, vídeo y/o los datos de los participantes en una conferencia multipunto. El MP puede procesar una única secuencia multimedia o varias simultáneamente, dependiente del tipo de conferencia soportada.

2.2.3.- Protocolos de señalización en H.323

Los protocolos de señalización más importantes utilizados en el seno de la H.323 son:

- **H.225.0**: Define la señalización entre terminales/gateways y gatekeeper (RAS). También define la señalización para establecimiento y liberación de la llamada (Setup, Alerting,...) que va por el canal de señalización de llamada. En este caso se utiliza un subconjunto de las funciones proporcionadas por la Q.931.
- **H.245**: Señalización de control extremo a extremo. La función principal es el intercambio de capacidades entre los terminales H.323 previa a la transmisión de información.
- **H.235**: trata sobre la seguridad en la comunicación incluyendo autentificación, autorización, control de llamada seguro y privacidad de los canales de voz
- **H.450**: señalización para el control de todos los servicios suplementarios (desvío de llamada, llamada en espera,...)

La arquitectura de protocolos utilizada por H.323 es la mostrada en la figura II.4.

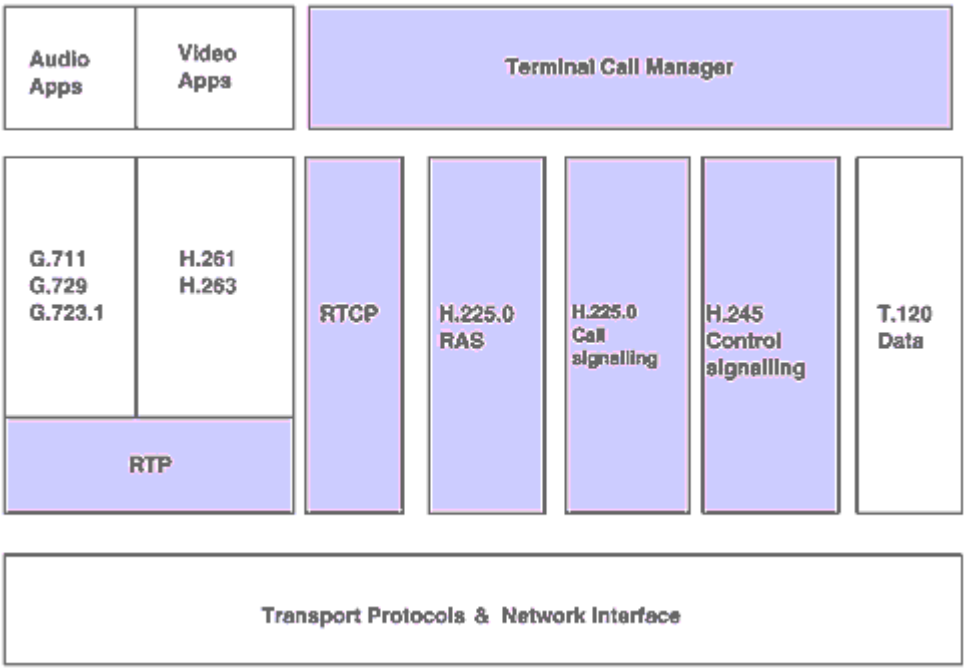


Figura II.4. Arquitectura de Protocolos en H.323

2.2.4.- Fases de una llamada H.323

Una llamada H.323 puede dividirse en tres fases en relación con los protocolos de señalización que intervienen en la misma:

- **RAS (Registro Autenticación y Estado)**: Cuando un terminal quiere hacer una llamada, pide permiso al gatekeeper mandando un paquete ARQ (Admission Request). Este mensaje contiene, entre otras cosas, los alias del destino (nombre o teléfono del usuario con el que quiere comunicarse). El GKR puede dar permiso para la llamada con un ACF (Admission Confirm) que contiene la dirección de transporte asociada al alias destino o su propia dirección de transporte si decide encaminar la señalización H.225.0. El GKR puede también denegar la llamada con un ARJ (Admission Reject) dando la razón por la cual la llamada no se ha cursado (por ejemplo, no hay suficiente ancho de banda). Durante esta fase el GKR realiza tres funciones: traducción de direcciones, autorización de llamada y gestión del ancho de banda.
- **H.225.0**: Es un subconjunto de mensajes del protocolo de señalización Q.931 de RDSI. Proporciona una conexión lógica entre los dos terminales. En las primeras versiones de la norma, H.225.0 se implementaba sobre TCP pero a partir de la versión 3 existe la posibilidad de utiliza UDP por problemas de retardo.
- **H.245**: Tan pronto como acaba la fase anterior, los dos terminales intercambian sus capacidades. Se ponen de acuerdo en el tipo de información que van a mandar.

Después de estas tres fases, se abren los canales lógicos entre los dos terminales de acuerdo con las capacidades intercambiadas y la comunicación multimedia comienza.

Los paquetes de audio y vídeo se transmiten sobre UDP, por lo que pueden desordenarse, perderse o retrasarse. Por esta razón se utiliza por encima de UDP el protocolo RTP (Real-Time Transport Protocol). Este protocolo se utiliza para permitir compensar el jitter y el desorden de los paquetes en recepción. El protocolo RTCP (RTP Control Protocol) se usa junto con RTP y permite cierta realimentación sobre la calidad recibida.

La señalización H.225.0 y H.245 puede encaminarse por el GKR o no en función de que se utilice el denominado modelo directo o indirecto. Si el GKR intercepta todos los mensajes de señalización puede realizar gestión de las llamadas manteniendo una tabla con las llamadas activas, el estado de los terminales, etc.

Por tanto, para establecer una comunicación H.323 se abren 3 canales de señalización (H.225.0, H.245 y RAS) más los canales lógicos de audio, vídeo y datos.

Uno de los mayores problemas de H.323 es elevado retardo de establecimiento de llamadas. Con objeto de mejorar la eficiencia, a partir de la versión 2 de la norma se reduce el tiempo de establecimiento de llamada con dos procedimientos: Fast Connect y encapsulado de mensajes H.245 en mensajes Q.931.

2.3.- SIP

2.3.1.- Introducción a SIP

El protocolo SIP cuyas siglas se corresponden con “Session Initation Protocol” es un estándar promovido por el IETF (Internet Engineeering Task Force) para las conferencias multimedia sobre IP. SIP es un protocolo de control de la capa de aplicación (definido en la RFC 2543) que puede ser utilizado para el establecimiento, mantenimiento y finalización de llamadas entre dos o más terminales (end-points). Como en otros protocolos VoIP, SIP esta diseñado para encargarse de las funciones de señalización y gestión de la sesión dentro de una red de telefonía por paquetes. La señalización permite el transporte de información de la llamada a lo largo de la red. Mientras, la gestión de la sesión proporciona la habilidad de controlar los atributos de una llamada extremo a extremo. SIP proporciona las capacidades de:

- Determinar la localización del terminal destino. SIP soporta resolución de direcciones, mapeo de nombres, y redireccionamiento de llamadas.
- Determinar las capacidades multimedia del terminal destino. Esto se realiza a través del protocolo SDP (Session Description Protocol), SIP determina el “nivel mínimo” de servicios común para los extremos. Las conferencias se establecen usando solamente las capacidades que puedan ser soportadas por ambos extremos.
- Determinar la disponibilidad del terminal destino. Si una llamada no puede ser realizada debido porque el destino no esta disponible, SIP determina si el conferenciante destino está todavía al teléfono o si aún no ha respondido en un número de reintentos adecuado. En dicho caso, devuelve un mensaje indicando por qué el destino no se encontraba disponible.
- Establecer una sesión entre los terminales origen y destino. Si la llamada puede ser completada, SIP establece una sesión entre estos puntos. SIP también soporta cambios en medio de la llamada, tales como la incorporación de un nuevo usuario a la conferencia o un cambio en las características multimedia, o incluso un cambio del codec.

- Manejar la transferencia y finalización de llamadas. SIP soporta la transferencia de llamadas desde un terminal a otro. Durante una transferencia de llamada, SIP simplemente establece una sesión entre el antiguo y el nuevo terminal (especificado por el conferenciante transferido) y termina la sesión entre el antiguo y el nuevo conferenciante . Al final de la llamada, SIP finaliza la sesión entre todos los participantes.

Las conferencias pueden estar constituidas por dos o más usuarios y pueden ser establecidas usando multicast o diferentes sesiones unicast.

2.3.2.- Componentes de SIP

SIP es un protocolo extremo a extremo. Los extremos en una sesión son llamados Agentes Usuarios (User Agents UAs). Un agente usuario puede funcionar en uno de los siguientes roles:

- Agente usuario cliente (UAC) – Una aplicación cliente que inicia la petición SIP.
- Agente usuario servidor (UAS) – Una aplicación servidor que contacta con el usuario cuando recibe una petición SIP y que devuelve una respuesta en representación del usuario.

Típicamente, un terminal SIP es capaz de funcionar en ambos sentidos como servidor y como cliente, pero funciona solo como uno u otro en cada comunicación. Si el terminal funciona como UAC o UAS depende del agente usuario que inicie la petición.

Desde el punto de vista de arquitectura, los componentes físicos de una red SIP pueden ser agrupados en dos categorías: clientes y servidores. En la figura II-5 se ilustra la arquitectura de una red SIP.

Adicionalmente, los servidores SIP pueden interactuar con otras aplicaciones, tales como servidores LDAP (Lightweight Directory Access Protocol), servidores de localización, bases de datos, o una aplicación XML (extensible markup language). Estas aplicaciones de servicios proporcionan servicios tales como directorio, autentificación, y tarificación.

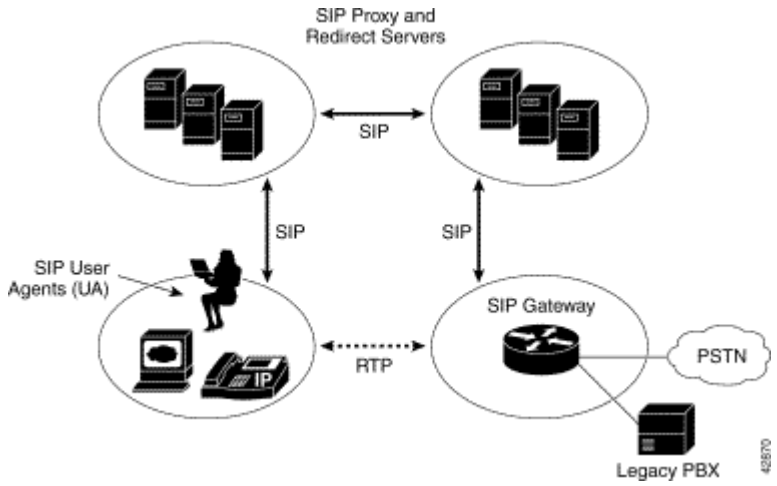


Figura II-5: Arquitectura SIP

Clientes SIP

Los clientes SIP incluyen:

- Teléfonos – Pueden actuar como UAS o UAC. Los teléfonos software (es decir, un ordenador que presenta capacidades telefónicas) y los teléfonos IP pueden iniciar las peticiones SIP y responder a estas.
- Pasarelas (Gateways) .- Proporcionan control de llamada. Las pasarelas habilitan diferentes servicios, donde el más común es la función de traducción entre terminales SIP y otro tipo de terminales. Esta función incluye la tradución entre diferentes formatos de transmisión y entre procedimientos de comunicación. Adicionalmente, la pasarela traduce entre diferentes codecs de audio y video, y lleva a cabo el inicio y finalicación de la llamada en el extremo LAN y el extremo de la red de circuitos conmutados.

Servidores SIP

Los servidores SIP incluyen:

- Servidor Proxy – El servidor proxy es un dispositivo intermedio que recibe peticiones SIP desde un cliente y entonces reenvía la petición en representación del cliente. Básicamente, los servidores proxy reciben mensajes SIP y los reenvían hacia el siguiente servidor SIP en la red. Los servidores proxy pueden

proporcionar funciones tales como autentificación, autorización, control de acceso a la red, rutado, peticiones de retransmisión de confianza, y seguridad.

- Servidor “Redirect” – Proporciona al clienteinformación sobre los siguiente pasos que un mensaje debería tomar y entonces el cliente contacta con el siguiente servidor o directamente con el UAS.
- Servidor “registrador” – Procesa las peticiones de los UACs para el registro de los actuales posiciones. Estos servidores son frecuentemente utilizados junto con un servidor proxy o un servidor “redirect”.

2.3.3.- Funcionamiento de SIP

SIP es un sencillo protocolo basado en ASCII que utilizas peticiones y respuestas para establecer la comunicación entre los diferentes componentes de la red y para establecer una conferencia entre dos o más terminales.

Los usuarios en una red SIP se identifican a través de una dirección SIP única. Una dirección SIP es similar a una dirección de correo electrónico y tiene el formato: userID@gateway.com. El identificador de usuario (userID) puede ser tanto un nombre de usuario como una dirección E.164.

Los usuarios se registran a través del servidor "registrador" usando su dirección SIP asignada. El servidor "registrador" proporciona esta información al servidor de localización tras una petición.

Cuando un usuario inicia una llamada, se envía una petición SIP a un servidor SIP (puede ser un proxy o un "redirect"). La petición incluye la dirección del llamante (en el campo de cabecera FROM) y la dirección del usuario destino (que aparece en el campo de cabecera To). En las siguiente secciones se incluyen diferentes ejemplos de llamadas extremo a extremo establecidas a través de un proxy y un servidor "redirect".

Con el paso del tiempo, un terminal SIP podría moverse entre diferentes sistemas finales. La localización del usuario puede ser registrado dinámicamente en el servidor SIP. El servidor de localización puede usar uno o más protocolos (incluyendo finger, rwhois y LDAP) para localizar al usuario final. Debido a que el usuario puede estar conectado a más de una máquina y debido a que el servidor de localización puede en alguna ocasión disponer de información imprecisa, podría devolver más de una dirección del usuario final. Si la petición se está procesando a través de un proxy, este servidor proxy probará cada una de las direcciones proporcionadas hasta localizar al usuario. Si la petición proviene de un servidor "redirect", este servidor reenvía todas las direcciones al usuario origen en el campo de cabecera "Contact" de la invitación de respuesta.

A. Utilización de una servidor Proxy

Si se utiliza un servidor proxy, el origen UA envía una petición INVITE al servidor proxy, este determina el camino, y enonces reenvía la petición al destinatario. Lo podemos apreciar en la figura II-6.

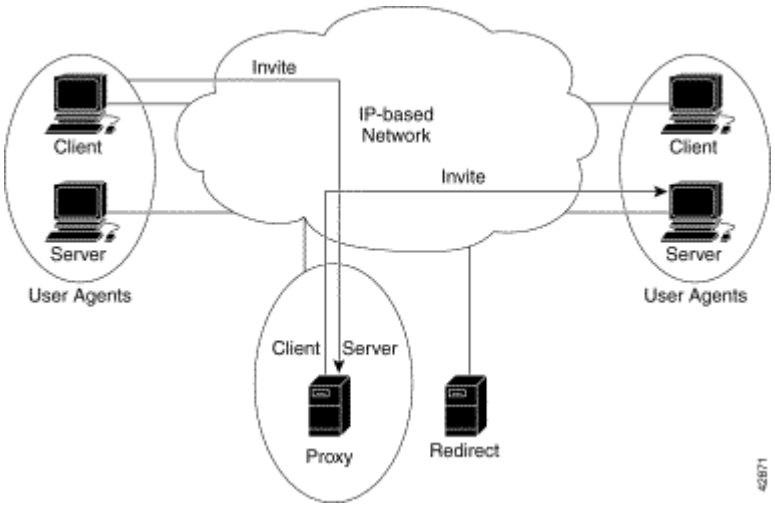


Figura II-6: Petición Sip a través de un proxy

El destinatario responde al servidor proxy, el cual posteriormente reenvía la respuesta al usuario origen de la llamada. Lo podemos ver en la figura II-7.

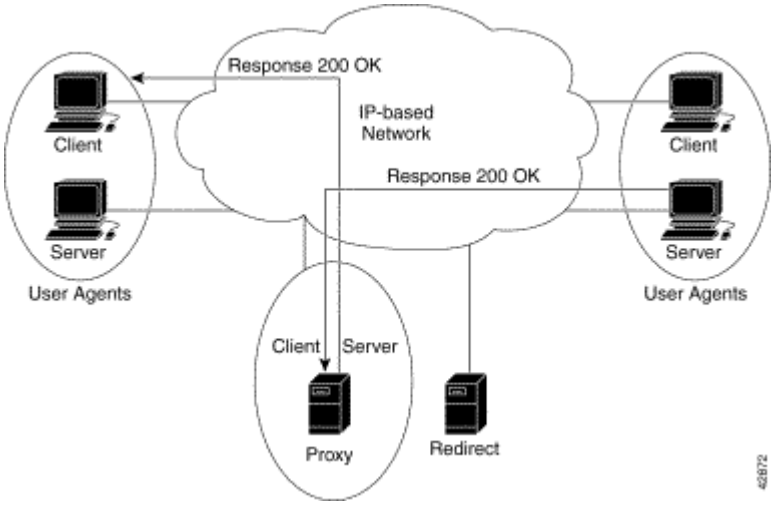


Figura II-7: Respuesta SIP a través de un proxy

El servidor proxy reenvía los ACKs de ambos participantes. Entonces una sesión es establecida entre origen y destino. El protocolo RTP (Real-time Transfer Protocol) se utiliza para la comunicación entre los extremos. Lo vemos en la figura II-8.

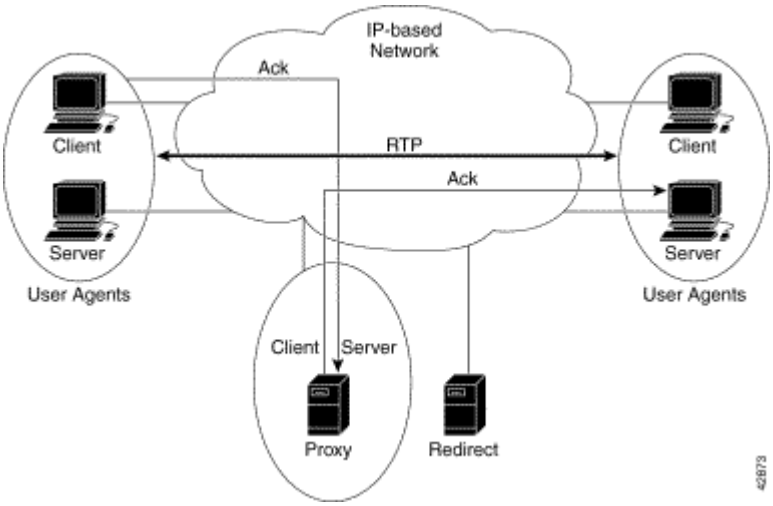


Figura II-8: Sesión SIP a través de un proxy

Usando un Servidor "redirect"

Si se utiliza un servidor "redirect", el terminal UA envía una petición INVITE al servidor "redirect", el cual contacta con el servidor de localización para determinar con es el camino hasta el destino, y entonces enviar esa información al usuario origen de la llamada. Este finalmente devuelve un ACK de la información. Podemos apreciar este proceso en la figura II-9.

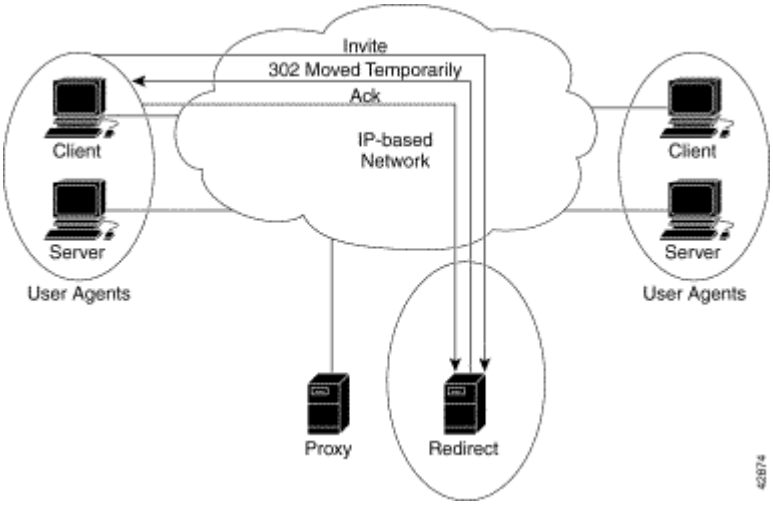


Figura II-9: Petición SIP a través de servidor "redirect"

El llamante envía entonces una petición al dispositivo indicado en la información de redirección (la cual podría ser el terminal destino final u otro servidor intermedio que reenviará la petición). Una vez que la petición alcance el terminal destino, este devuelve una respuesta y el origen mandará un ACK de esta respuesta. El protocolo RTP se utiliza para la comunicación entre origen y destino. Veámoslo en la siguiente figura II-10.

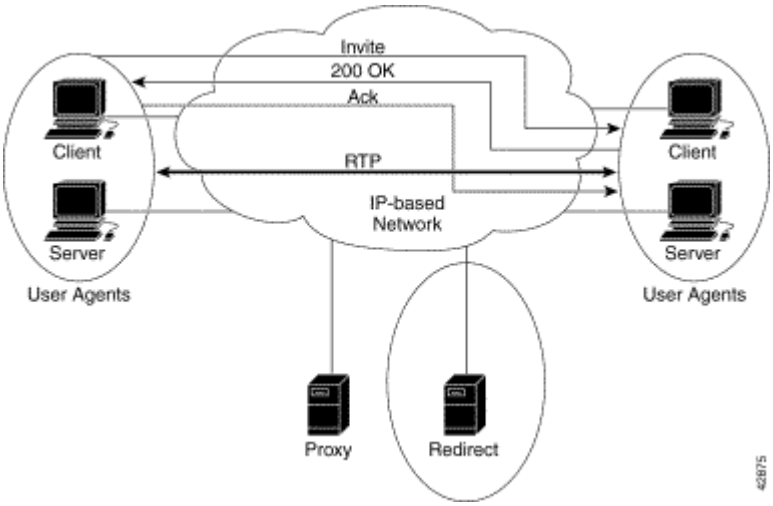


Figura II-10: Sesión SIP a través de un servidor “redirect”

2.3.4.- SIP Versus H.323

SIP y H.323 fueron diseñados para encargarse del control de sesión y las funciones de señalización en una arquitectura de control de llamada distribuida. A pesar de que SIP y H.323 pueden ser usados también para comunicar a terminales de inteligencia limitada, están especialmente bien preparados para la comunicación con terminales inteligentes.

En la tabla II-2 podemos apreciar una breve comparativa entre SIP y H.323.

Aspecto	**SIP**	**H.323**
Clientes	Inteligentes	Inteligentes
Inteligencia de red y servicios	Proporcionada por los servidores (Proxy, Redirect, Registrador)	Proporcionada por los gatekeepers
Modelo usado	Internet / WWW	Telefonía / Q.SIG
Protocolo de señalización	UDP o TCP	TCP (UDP es opcional en la versión 3)
Protocolo multimedia	RTP	RTP
Codificación de base	ASCII	Binario (codificación ASN.1)
Otros protocolos utilizados	Protocolos IETF/IP, tales como SDP, HTTP/1.1, IPmc, y MIME	Protocolos ITU / ISDN, tales como H.225, H.245, y H.450
Interoperabilidad entre vendedores	Ampliamente extendida	Limitada

Tabla II.2: SIP versus H.323

A pesar de que los mensajes SIP no son directamente compatibles con H.323, ambos protocolos pueden coexistir en la misma red de telefonía por paquetes si un dispositivo que permita la interoperabilidad se encuentra disponible. Entonces, después de que la comunicación se haya establecido, la información de setup se transmite entre las diferentes pasarelas como un flujo RTP.

2.4.- MGCP / MEGACO / H.248

MGCP (Media Gateway Control Protocol) es una definición de comandos que debería usar un servidor de telefonía para controlar el interfaz entre una RTC y una red de VoIP. MGCP incluye también una definición de señales que el dispositivo de interfaz debería devolver al servidor. Un servidor de telefonía consiste en un sistema de dispositivos que implementa alguna aplicación telefónica como pueda ser un conmutador, una centralita, o una pequeña compañía telefónica. Los comandos y señales de MGCP se definen como paquetes IP, que permiten la independencia con el sistema operativo y el lenguaje de programación. Con MGCP, un agente (también llamado Media Gateway Controller) puede correr en un plataforma de propósito general. El agente puede mandar comandos a las pasarelas para realizar llamadas o terminarlas, tanto en el lado de RTC como en el lado IP.

MGCP versus SIP/H.323

Suele existir confusión sobre la utilización de MGCP vs. SIP o H.323. MGCP es perfecto para construir sistemas grandes y escalables que pueden interoperar con SIP y H.323, pero no es un recambio para SIP o H.323.

MGCP no es un protocolo para la inicialización y liberación de llamadas, como SIP y H.323. Con estos últimos cada dispositivo es un terminal. Un dispositivo puede iniciar una llamada con otro dispositivo, y viceversa. Con MGCP, un servidor de telefonía, denominado agente o “Media Gateway Controller” inicializa las llamadas entre diferentes pasarelas.

Con H.323, por ejemplo, cuando se establece una llamada en una pasarela, esta junta los dígitos del número marcado, y entonces decide donde debe rutar la llamada. Entonces la pasarela H.323 contacta con el terminal H.323 y comienza la llamada. Previamente se realiza el proceso de intercambio de capacidades para determinar las características de la llamada. Entonces se establece una sesión RTP/RTCP que será la encargada de transmitir el audio o video entre ambos extremos.

Por el contrario, con MGCP cuando se establece una llamada en una pasarela, esta junto los dígitos del número marcado y se los pasa al agente. La pasarela no hace nada más, ni intercambio de capacidades, ni determinar dónde está el terminal. Será misión del agente determinar esto antes de establecer la llamada. El agente podría usar H.323 para negociar las especificaciones de la llamada con los terminales, y usar MGCP para establecer la sesión RTP/RTCP con la pasarela que tiene la llamada entrante. El agente podría también utilizar SIP o SS/ o alguna combinación de estos protocolos para establecer la llamada con otras pasarelas.

Hemos visto como H.323 y SIP difieren en el método de establecimiento y control de la llamada. Y aquí es donde encaja MGCP. Este se preocupa estrictamente del establecimiento de la sesión entre las sesiones RTP y RTC, y deja la inicialización y control al agente. De alguna manera MGCP es un unificador, ya que sería posible crear un agente que aceptase llamadas SIP y H.323 y se utilizaría MGCP para establecer la llamada entre la pasarela MGCP y los terminales o pasarelas H.323 y SIP.

Veamos la evolución de estos protocolos de control. En un principio ETSI (TIPHON) propuso una arquitectura distribuida para la implementación de pasarelas que está basada en tres componentes – Media Gateway, Media Gateway Controller y la Signalling Gateway. Todos los cuerpos de estandarización han aceptado esta arquitectura y tanto la ITU-T como el IETF han estado trabajando en la definición de la interfaz entre las tres componentes de la pasarela.

Tanto la ITU-T SG16 como IETF WG MEGACO han estado estudiando el interfaz entre la Media Gateway (MG) y la Media Gateway Controller (MGC) para soportar la comunicación MG-MGC. La IETF inicialmente definió la interfaz en la especificación MGCP y más tarde propuso una nueva definición de interfaz en la especificación MEGACO. La ITU-T de forma paralela publicó su propia especificación dando lugar al protocolo H.248 (también llamado H/GCP).

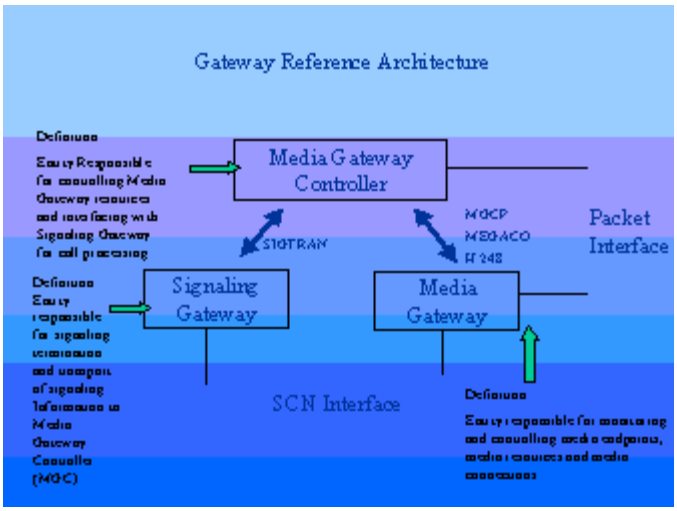


Figura II.11 – Arquitectura

En primer lugar veremos la necesidad y evolución de protocolos de control de pasarelas y después discutiremos las diferencias entre MEGACO en su versión 0.5 y el protocolo MGCP versión 1.0.

La necesidad de un protocolo para el control de pasarelas

La descompuesta arquitectura de pasarelas descrita anteriormente distribuye la funcionalidad de control de llamadas y procesamiento multimedia sobre diferentes elementos de la red, la Media Gateway Controller y la Media Gateway respectivamente. En consecuencia, aparece la necesidad de un protocolo de control entre estas entidades que permita al control de llamada inicializar las conexiones del medio y las propiedades de llamada basados en las necesidades de comunicación. A continuación se resumen los requisitos funcionales del protocolo de control de pasarelas.

Control de recursos

El protocolo de control de pasarelas permite al MGC reservar y anular a los terminales y los recursos multimedia para el uso de una llamada particular.

- El protocolo proporciona la flexibilidad que permite a la MGC especificar los recursos requeridos para una llamada o permite que la MG seleccione de una pila de recursos e informe a la MGC.

- El protocolo de control permite a la MGC obtener el estado de los recursos en la MG.
- Gestión de conexiones.
- El protocolo de control de pasarelas permite a la MGC crear conexiones tanto de paquetes como de circuitos en cualquier combinación. Las terminaciones deberán ser TDM, o análogo, Ethernet, ATM o Frame Relay.
- El protocolo soporta el establecimiento de flujos unidireccionales, simétricos bidireccionales, asimétricos bidireccionales, punto a punto y punto-multipunto, sobre diferentes tipos multimedia como audio, texto, video, etc.
- El protocolo permite que la MGC añada o elimine uno o más flujos multimedia a una conexión según se requiera durante la llamada.

Control de procesamiento multimedia

El protocolo de control de pasarelas permite a la MGC especificar la variación de parámetros del medio para cada flujo que forme parte de la llamada. Esto incluye modificaciones tales como la adaptación de flujos entre diferentes tipos de transporte.

- El protocolo permite a la MGC especificar procesamiento multimedia especial como podría ser la cancelación de eco, detección de tono, supresión de silencio.
- El protocolo habilita a la MGC para añadir multimedia, como podría ser la reproducción de anuncios, o la extracción multimedia como la detección DTMF, terminación MODEM, fax, etc.

Procesamiento de eventos y señales

- El protocolo de control de pasarelas permite a la MGC especificar los eventos que se desean monitorizar o las señales que serán aplicadas en la MG sobre un flujo en concreto de una llamada.
- El protocolo proporciona mecanismos para notificar los eventos detectados por la MG a la MGC.

- El protocolo permite a la MGC especificar las acciones (p.ej. notificar eventos a la MGC, aplicar otra señal, acumular eventos hasta que sean solicitados, etc.) que serán llevadas a cabo en la MG cuando tenga lugar un evento. De igual manera, permite la MGC especificar cuando una señal aplicada a un flujo debe ser eliminada (p.ej. después de un tiempo, cuando suceda un evento, cuando se solicite otra señal, etc.)

Notificación de estadísticas

- El protocolo de control de pasarelas habilita un mecanismo que permite a la MG la notificación de estadísticas tales como estadísticas de calidad de servicio, duración de una llamada, volumen de contenido, etc. Recogidas durante una llamada.
- El protocolo soporta un mecanismo por el cual la MGC puede notificar estas estadísticas en cualquier momento durante una llamada.

Gestión de conexiones

- El protocolo de control de pasarelasa permite el establecimiento de una conexión de control entre MGC y MG.
- Permite a una MGC tener conexiones a múltiples MGs y viceversa, por ejemplo, es posible que múltiples MGCs controlen una sola MG.
- Estos términos engloban diseño, planificación, aprovisionamiento, mantenimiento, rendimiento, seguridad, contabilidad y peticiones de clientes y control de gestión de telecomunicaciones.

Transporte

- El protocolo de control de pasarelas proporciona un mecanismo de transporte fiable para el intercambio de mensajes entre MG y MGC. El mecanismo de transporte permite la detección de fallos en el transporte y soporta un gran número de conexiones de control.
- El mecanismo de transporte proporciona un mecanismo para el intercambio de comandos y respuestas entre entidades así como la detección y eliminación de comandos y respuestas duplicados.

Seguridad

El protocolo de control de pasarelas permite comunicaciones seguras entre la MG y la MGC. Permite autentificación mutua entre MG y MGC, protección sobre el intercambio de mensajes de control entre las dos entidades y atenúa los ataques por denegación de servicios.

Soporte de aplicaciones

El protocolo permite a la MGC proporcionar servicios como faz en tiempo real o conferencia usando los recursos de procesamiento de señal disponibles en la MG.

Evolución de los protocolos de control de pasarelas

Las especificaciones de protocolos del IPDC (IP Device Control Protocol) y el SGCP (Simple Gateway Control Protocol) fueron los primeros candidatos que compitieron por el protocolo de control de pasarelas descrito anteriormente. MGCP, MEGACO y H.248 (llamado anteriormente H.GCP) son los sucesores de estos protocolos. Todos definen la interfaz entre la MG y la MGC identificadas en la arquitectura de pasarelas distribuidas propuesta por el ETSI-TIPHON. La evolución de estos estándares ha captado la atención de la industria en nuestros días.

- El protocolo MGCP aparece a partir de la fusión de los protocolos SGCP e IPDC.
- El grupo de trabajo de MEGACO de la IETF (aprobado por la IESG en Enero de 1999), responsable de la estandarización del interfaz de control entre MG y MGC adoptó MGCP en su versión 0.1 como la primera solución.
- El grupo de MEGACO trabajo en la evolución del protocolo MGCP hasta la revisión 3 de este protocolo, llegando su release en Febrero de 1999, pero abandonó el proyecto debido a la mayor aceptación de otros protocolos (MDCP) de la ITU-T.
- Mientras, la evolución de MGCP continuó y finalmente se convirtió en la RFC 2705 en Octubre de 1999 después de la quinto revisión de su draft.

- Entonces el grupo de trabajo de MEGACO empezó a trabajo en un protocolo de compromiso entre MGCP y MDCP, que posteriormente se denominó protocolo MEGACO. El primer drft de MEGACO apareció en Marzo de 1999. En paralelo a la IETF, la ITU-T estaba evaluando diferente opciones y en Abril de 1999 la ITU-T SG16 adopto MEGACO en su versión 0.1 como la especificación para comenzar para el protocolo ITU-T, que se llamó inicialmente H.GCP y posteriormente pasó a denominarse H.248.
- El ITU-T SG16 introdujo contexto multimedia en el protocolo en Mayo/Junio y el grupo de trabajo del IETF comenzó un debato sobre su adopción o rechazo.
- Finalmente se decidió mejorar el protocolo para habilitar el soporte multimedia. Se alcanzó un acuerdo en Junio de 1999 entre el IETF y la ITU-T para continuar con un único documento del protocolo. En consecuencia todos las siguientes revisiones del protocolo fueron las mismas para el IETF y la ITU-T.
- Las reuniones del IETF en Oslo y Washington, las reuniones de la ITU-T en Berlín y una gran actividad en la lista de correo de MEGACO resolvieron diferentes problemas referentes al protocolo.

2.5.- Calidad de servicio

Hasta el momento no hemos profundizado en la calidad que podemos obtener en una comunicación de Voz sobre IP. Como hemos comentado a lo largo de este capítulo, una de las principales diferencias con la telefonía tradicional es que esta se venía realizando a través de conmutación de circuitos, mientras que VoIP funciona sobre redes de paquetes, es decir, por conmutación de paquetes.

Esta nueva característica en la telefonía IP tiene importantes implicaciones en la calidad que podemos esperar en una comunicación de este tipo. Debemos apreciar que cada paquete puede viajar por diferentes caminos, ofrecer retardos diferentes, o incluso debemos aceptar la pérdida de paquetes.

En consecuencia aparecen una serie de problemas para esta comunicación, algunos directamente relacionados con la red en que estamos trabajando y otros derivado de las características especiales de un tráfico de audio. Veamos una lista de estos problemas:

Problemas derivados de las características de la red:

- Retardo en la red
- Variación del retardo (jitter)
- Posibilidad de la pérdida de paquetes
- Escaso ancho de banda

Problemas derivados de las características del audio:

- Aparición de ruido
- Ganancia del sistema
- Posible existencia de eco
- Pérdidas en los transductores (micrófono y/o altavoces)
- Nivel de ganancia del sistema

Deberemos tener en cuenta todos estos parámetros cuando deseemos evaluar la calidad de una comunicación basada en VoIP. Algunas de estas medidas nos determinarán si la comunicación podrá ser establecida con unas garantías mínimas de calidad. El ejemplo más claro lo podemos ver con el retardo de transmisión en la red IP. Si este valor supera los 0,5 segundos, la comunicación es similar a las antiguas comunicaciones transoceánicas donde la falta de realimentación por parte del receptor resulta en una comunicación torpe y no fluida.

Para comprender como afectan estos parámetros a una comunicación y estudiar como pueden ser atajados analicemos en mayor profundidad como funciona un sistema VoIP extremo a extremo.

En primer lugar cabe destacar que para transportar la voz sobre IP las muestras audio se digitalizan, se codifican y se transmiten en pequeños paquetes numerados. Por el lado del receptor estos paquetes se almacenan en un buffer y se reproducen posteriormente.

Veamos este proceso en un gráfico:

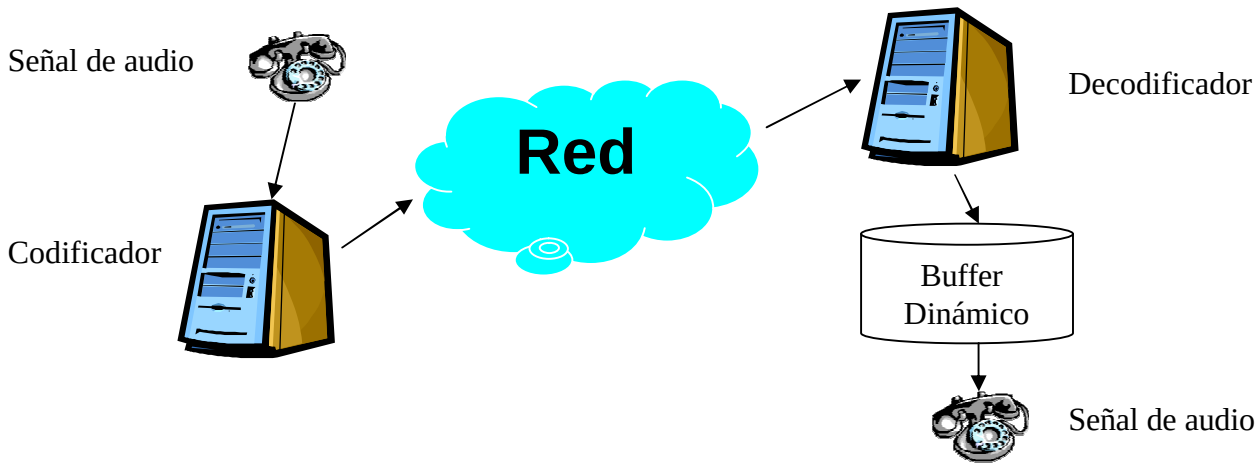


Figura II-12: Ejemplo de una comunicación VoIP

Veamos uno a uno los diferentes factores que pueden influir en las calidad de la comunicación en una sesión VoIP:

- Codificadores:

Como hemos venido comentando a lo largo de este capítulo, los codecs constituyen uno de los puntos más importantes de la telefonía IP. Si bien la característica más importante de esta telefonía es la digitalización y paquetización del audio, la posterior compresión proporcionada a través de los codecs nos permite hacer un uso eficiente del ancho de banda disponible.

Sin embargo, la mayor o menor capacidad de los codecs incide directamente sobre la calidad de la comunicación, ya que la mayoría de estos codecs implementan algoritmos de compresión con pérdidas. En consecuencia será interesante seleccionar el codec adecuado según cada situación para llegar a un compromiso entre el ancho de banda consumido y la calidad obtenida.

En primer lugar debemos nombrar los dos codecs utilizados en la telefonía tradicional. Se trata de G711 en sus dos variantes ley u y ley A. Estos codecs muestrean la señal de audio a 8KHz y codifican con 8 bits, alcanzando de esta manera una tasa de 64 Kbit/s. La calidad obtenida es muy buena, si bien el ancho de banda utilizado es bastante elevado.

Veamos las características de otra serie de codecs más interesantes:

Codec	G723.1	G729	G729A
Ancho de Banda	5.3 / 6.4 kbps	8 kbps	8 kbps
Tamaño de trama	30 ms	10 ms	10 ms
Requerimiento CPU	16 mips	20 mips	10.5 mips
Retardo de procesamiento	30 ms	10 ms	10 ms

Tabla II-3: Características de los codecs más importantes

El ancho de banda indicado en la tabla hace referencia al régimen binario de salida cuando a la entrada colocamos una señal PCM con 64 kb/s, es decir, una señal muestreada a 8 KHz y utilizando 8 bits por muestra. El requerimiento de CPU indica la velocidad de procesamiento mínima necesaria medida en millones de instrucciones por segundo.

A tenor de este esquema, podemos apreciar que G723.1 es el codec con menor requisito de ancho de banda, de manera que este podría resultar el más interesante para usuarios residenciales que típicamente no disponen de una tasa binaria muy elevada.

Para entornos empresariales donde el ancho de banda disponible puede ser más elevado resultaría más interesante G729 que ofrece una mayor calidad de comunicación a partir de un retardo de procesamiento más bajo.

- Ancho de banda:

Resulta evidente la importancia de disponer de un ancho de banda adecuado para lograr una comunicación adecuada. Sin embargo como hemos podido ver en el apartado sobre codificadores, las necesidades de ancho de banda no son extremadamente elevadas. Actualmente cualquier acceso a Internet proporciona bastante más de los 8 kbps requeridos por los codecs con mayores requisitos. Sin embargo el problema deriva de otros parámetros ligados al ancho de banda como son el retardo y la variación de retardo.

- Retardo:

Este puede resultar uno de los parámetros más interesantes para determinar la viabilidad de una comunicación VoIP, de manera que intentaremos analizar de donde proviene exactamente. Cada uno de los componentes que aparecían en la figura II-1 introducen un retardo que debemos sumar para calcular el retardo final. Algunos de estos retardos serán fijos, como los que introducen los codificadores de voz, si bien existen otros variables como pueda ser el retardo debido a la red.

Veamos pues los diferentes elementos que introducen retardo:

- Codificadores: su retardo es inevitable y fijo. En la tabla II-1 pudimos ver los valores que pueden alcanzar. Estos valores nunca superarán los 30 mseg. de manera que su importancia no será muy radical.
- Transporte en de la red: se trata de un valor no determinista por naturaleza que depende de las características de la red. Su valor suele ser lo suficientemente importante como para considerarlo el más importante en la suma que estamos realizando.

  Debemos distinguir dos casos:
  - En las comunicaciones dentro de una red de área local (LAN) este valor será lo suficientemente bajo como para que no afecte a nuestra comunicación.
  - En el caso de comunicaciones entre puntos más distantes, el retardo provocado por el transporte de en la red dependerá del tipo de acceso disponible, así como del proveedor de servicios a Internet, y resultaría interesante realizar un estudio para comprobar la viabilidad de la comunicación.

- Variación del retardo de transporte: este valor se suele denominar “jitter” y se corresponde con la variación que pueda sufrir el retardo de transporte. Es decir, lo interesante en este caso no es la magnitud de este valor, sino la amplitud de las variaciones entre cada paquete.

  El valor del “jitter” nos determinará el tamaño del buffer de recepción que debemos colocar. Si este valor fuese lo suficientemente pequeño podríamos reproducir los paquetes inmediatamente una vez que sean recibidos, sin embargo si este valor es considerable será necesario que el tamaño del buffer sea lo suficientemente grande como para no perder paquetes que puedan llegar con un ligero retraso. Debemos tener en cuenta que el tamaño de este buffer determinará un nuevo retardo que sufrirá el audio antes de ser reproducido, de manera que debemos llegar a un compromiso.

  Al igual que comentábamos anteriormente, en redes LAN podemos disminuir el tamaño de este buffer tanto como deseemos, ya que en una red local el “jitter” se

puede considerar prácticamente nulo. Sin embargo en comunicaciones a larga distancia la utilización de una buffer de recepción adecuado puede mejorar notablemente la calidad de la comunicación.

Existe un hecho que acentúa aún más la importancia del retardo. Algunas arquitecturas de VoIP nos permiten incluir más de una trama de voz en cada paquete de datos, evitando de esta manera una sobrecarga de cabeceras y disminuyendo el ancho de banda utilizado. Sin embargo, de esta manera aumentamos el tiempo medio entre paquetes, obligándonos a utilizar un buffer de recepción mayor.

- Pérdida de paquetes:

Evidentemente las comunicaciones VoIP utilizan una red IP para el transporte de paquetes, la cual no asegura que los paquetes alcancen su destino. Existiría la posibilidad de utilizar un protocolo de transporte fiable tal como TCP, pero debido a que trabajamos con tráfico en tiempo real, no tiene ningún sentido utilizar este protocolo.

Este problema se soluciona parcialmente a través de la utilización de determinados codecs. Por ejemplo, el codec G.723.1 utiliza técnicas de interpolación para simular paquetes perdidos. En cualquier caso, mientras el porcentaje de perdidas sea lo suficientemente bajo, la calidad de la comunicación se mantendrá en unos niveles aceptables.

Analizando todos estos factores podemos apreciar claramente que en el caso de que la red presente una congestión de relativa intensidad, la pérdida de paquetes, y en mayor medida, el retardo de transmisión y la variación de este provocará que la comunicación sea impracticable.

Adicionalmente los terminales deben tener la suficiente potencia para soportar los codificadores precisos. Esta características no resulta muy importante debido al

conocido ritmo de crecimiento de velocidad de los procesadores, y a que cualquier Pentium puede funcionar perfectamente como terminal.

Con respecto al eco debemos indicar que existen diferentes tipos de este. En primer lugar tenemos el eco debido a la conversión de una configuración de 4 hilos a otra de 2 hilos. En el caso de comunicaciones PC-PC este eco no existe. Por otro lado puede aparecer eco debido a la utilización de altavoces y micrófonos típicos de un entorno multimedia en lugar de un teléfono tradicional. Para eliminar este tipo de eco es necesario configurar adecuadamente los niveles de ganancia de los altavoces y del micrófono.

Mecanismo de reserva de ancho de banda

Una vez expuestos todos los parámetros que afectan a la calidad de la señal, y conociendo las características de acceso típicas que puede conseguir un usuario residencial o en una empresa, que se encuentra caracterizado por un ancho de banda pequeño y bruscas variaciones del jitter, deberíamos preguntarnos hasta que punto puede resultar interesante la implementación de esta nueva tecnología.

Por una parte ya hemos comentado que su uso en redes de área local permite una calidad de comunicación muy elevada. De manera que podríamos pensar en un entorno de trabajo de una empresa para la realización de llamadas internas sin la necesidad del uso de una centralita telefónica.

Pero por otro lado nos encontramos las comunicaciones entre usuarios residenciales a nivel provincial, interprovincial, e incluso, al extranjero, o en el entorno de la empresa para llamadas al exterior. En estos casos, las actuales redes de paquetes no son suficientes para asegurar una calidad adecuada. Sin embargo existen diferentes mecanismos que nos pueden permitir asegurar cierta calidad de servicio.

Hay un debate importante sobre si son necesarios determinados mecanismos para garantizar las condiciones deseadas de calidad del servicio, o bien si por contra la evolución tecnológica natural de las redes, especialmente con la multiplexación por

división de longitud de onda, hará que la oferta de anchura de banda sea tan abundante que resulte innecesario acudir a otro tipo de medidas. Otras opiniones consideran que aunque la capacidad de las redes mejore surgirán nuevas aplicaciones cada más exigentes en términos de anchura de banda. Sea cual sea la respuesta acertada a largo plazo, es cierto que durante un período de tiempo todavía apreciable, la garantía de unas condiciones determinadas de calidad del servicio (QoS) va a exigir algún tipo de mecanismo, ya que las condiciones de capacidad distarán de las óptimas en términos de adecuación de la oferta a la demanda.

La IETF, Internet Engineering Task Force, ha propuesto varios modelos de servicio y mecanismos para satisfacer la demanda de QoS. Los más conocidos son el modelo de servicios integrados / RSVP (Resource Reservation Protocol), el modelo DS de servicios diferenciados, la técnica conocida como MPLS (Multiprotocol Label Switching), la ingeniería de tráfico y el enrutamiento sujeto a restricciones.

El modelo de servicios integrados se caracteriza por la reserva de recursos de red. por ejemplo, en el caso de aplicaciones en tiempo real, antes de que se transmitan los datos, el protocolo establecerá los caminos y reservará los recursos necesarios. RSVP es un protocolo de señalización que permite realizar las funciones anteriores, en beneficio de la calidad de servicio de los tráficos encaminados. En el caso de los servicios diferenciados, los paquetes se marcan de forma diferente, creando diferentes clases que a su vez reciben distintas categorías de servicios. MPLS es un esquema de retransmisión, que asigna etiquetas a los paquetes en función de su prioridad de despacho. La ingeniería de tráfico está constituida por un conjunto de técnicas que tienen como objetivo organizar los procesos mediante los cuales los paquetes fluyen sobre la red. Finalmente, el enrutamiento bajo restricciones tiene en cuenta circunstancias como la anchura de banda disponible o el retraso predecible en la red.

Hay un gran número de artículos y publicaciones técnicas en los que se describen los méritos y las debilidades relativas de cada una de las aproximaciones anteriores. Una visión panorámica de la problemática emergente de la calidad de servicio en Internet podría resumirse en los términos siguientes:

- Los usuarios y los clientes han de negociar condiciones SLA (Service Level Agreement) con los proveedores de servicio Internet (ISP). Los SLA

especificarán los servicios y las calidades que los clientes recibirán, bien sean estáticos o dinámicos. En los acuerdos estáticos, los usuarios podrán transmitir / recibir información en cualquier momento, mientras que en los SLA dinámicos utilizarán un protocolo de señalización tipo RSVP para solicitar servicios y recursos bajo demanda, antes de la transmisión. Los brokers de anchura de banda en los dominios de los clientes deciden la forma en que se comparten los recursos disponibles en función de las condiciones de cada cliente. Los campos DS de los paquetes se marcan en consecuencia, para indicar los servicios deseados.

- Los routers de entrada de los ISP se configuran siguiendo reglas definidas, y los de salida también de forma consecuente. Las reglas pueden configurarse por el administrador de la red, o de forma dinámica mediante protocolos tales como LDAP o RSVP. Los ISP habrán de implementar algún tipo de control de admisiones con el fin de soportar SLA dinámicos.
- Con los MPLS se establecen label switched paths (LSP) entre cada par de ingresos / salidas del dominio en cuestión. En los routers de entrada de los ISP los campos denotativos de la clase de servicio (COS) se determinan a través de los resultados de la clasificación y del routing. Los encabezamientos MPLS se insertan entonces en los paquetes. Los routers internos procesan los paquetes según sus etiquetas y campos COS, que se elimina cuando los paquetes salen del dominio MPLS correspondiente.
- El routing sujeto a restricciones puede utilizarse para computar las rutas sometidas a condiciones y normas específicas de QoS. El objetivo en este caso es cumplir los requerimientos de calidad del servicio del tráfico y mejorar la utilización de las redes.
- MPLS y routing bajo restricciones son dos técnicas que pueden utilizarse conjuntamente para controlar el camino de los tráficos, evitar congestiones y mejorar el grado de utilización de las redes.
- No debe olvidarse que cada una de estas técnicas se sitúa en diversos niveles del modelo de referencia. MPLS está situada entre los niveles de enlace y red, mientras que el routing bajo limitaciones se encuentra en el nivel de red. RSVP y DS sin embargo actúan en el nivel transporte. Esta circunstancia marca, como es lógico, algunas diferencias significativas.

Capítulo 3 – Desarrollo de una plataforma VoIP[1]

Existen diferentes tecnologías que resultarían interesantes para el desarrollo de esta plataforma. En primer lugar podríamos optar por un entorno H.323, al tratarse de un estándar con mayor cantidad de equipos en el mercado o fabricantes que han apostado por su desarrollo. De igual forma existirían más proyectos de investigación que podrían resultar de interés para la creación de la plataforma.

Por otra parte aparecería un entorno SIP, que más tarde ha aparecido como una buena alternativa a H.323. Se trata de un protocolo algo menos recargado y por el que diferentes operadores de telefonía móvil han apostado muy fuerte.

El mayor auge y expectación del entorno H.323 en la época en que se creó la plataforma nos hizo inclinarnos por este en lugar de SIP. Sin embargo, el tiempo esta demostrando que en el futuro convivirán ambos protocolos, y aparecerán otro tipo de problemas a resolver como es la interoperabilidad entre equipos.

[1] En el presente capítulo, por motivos de seguridad, se han eliminado o substituido las direcciones IP de los equipos presentes por direcciones privadas.

3.1.- Creación de una plataforma VoIP

En primer lugar debemos indicar que el protocolo elegido inicialmente para la comunicación de audio en nuestra plataforma sería H.323. Este fue elegido debido principalmente a su amplia estandarización entre los fabricantes de clientes H.323. Tal es el caso del cliente Netmeeting que viene de la mano de Microsoft.

Posteriormente se propuso integrar esta plataforma con otra donde apareciese el protocolo SIP. Este último, algo menos recargado, fue el elegido en el entorno del proyecto Mobydick, de manera que se optó por intentar integrar ambas plataformas.

Una vez elegido el protocolo que debía implementarse en la plataforma, se prefijaban una serie de elementos que compondrían a la misma, de manera que se requería un estudio en profundidad de cada uno de estos, para su instalación y correcta configuración.

De esta manera los elementos que requerirían un estudio serían:

- Clientes H.323
- Gateway
- Gatekeeper

Para completar la creación de la plataforma se generarían una serie de manuales que permitiesen una instalación apropiada de estos elementos, y la configuración adecuada para lograr obtener los mejores resultados de calidad en la comunicación.

Si bien en un principio la inclusión del gatekeeper se desestimo esperando la maduración de la plataforma. Una vez se logró un escenario estable de trabajo, se decidió proceder a la instalación del gatekeeper que permitiría servicios de valor añadido como: la función de directorio, el control de acceso, el soporte de servicios de Help Desk o grupo de salto, la coordinación con mecanismos de calidad de servicio y el desarrollo de pasarelas de buzón de voz-email.

De esta forma, la plataforma finalmente ofrecería la posibilidad de una comunicación interna en el Departamento de Ingeniería Telemática a través de los diferentes clientes H.323, así como acceso a de Red Telefónica Conmutada (RTC) utilizando para ello los servicios de la plataforma. En última instancia la posibilidad de acceso a RTC se vería limitada a la comunicación con la centralita de la universidad para acceder a números internos.

3.1.1.- Descripción de la plataforma

Como hemos comentado, pretendemos ofrecer un servicio de telefonía en el Departamento de Ingeniería Telemática usando la tecnología de Voz sobre IP. Para ello tendremos que instalar una serie de elementos característicos de una red VoIP que comentaremos a continuación. Podemos ver en la figura III-1 una representación del escenario que deseamos montar. Posteriormente podremos ver los equipos que finalmente formarán la plataforma.

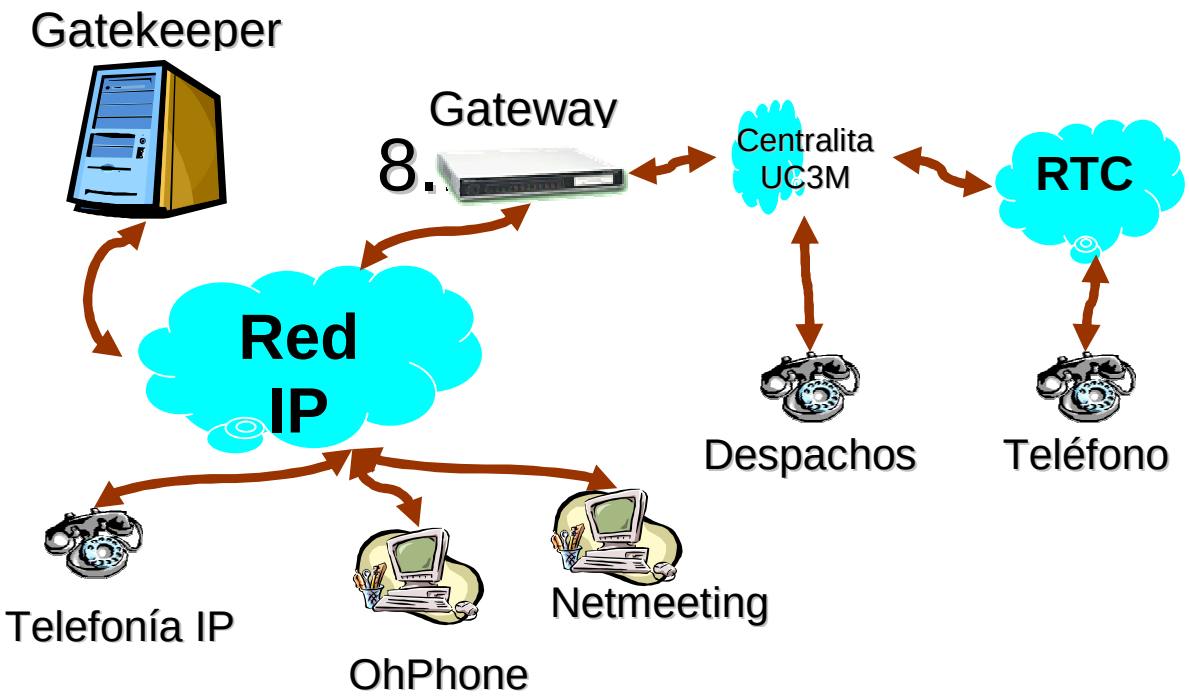


Figura III.1 – Modelo de la plataforma

- ***Clientes H.323***: necesitamos una serie de clientes que nos permitan acceder al servicio que pretendemos ofrecer. Estos clientes deberían disponer de una interfaz amigable para el usuario, y que permita una rápida comunicación sin necesidad de configuraciones especiales para cada usuario. Es decir, el administrador del sistema montará los clientes que consideré oportunos y los configurará para que el acceso resulte lo más sencillo posible.

  Entre los diferentes clientes por los que se puede optar aparecen dos como claros aspirantes:

  1. ***Microsoft Netmeeting 3.01***: se trata de la última versión del famoso cliente H.323 de la familia Microsoft. Tal y como deseábamos, Netmeeting ofrece una interfaz amigable, y un entorno de trabajo muy intuitivo. Hemos podido apreciar algunos problemas de compatibilidad con la Gateway con la que trabajaremos, pero las ventajas que ofrece son muy interesantes.

  2. ***OpenPhone 1.1 para Linux***: se trata de una opción muy interesante que nos ofrece mayor capacidad de configuración, lo que nos permite mejorar las prestaciones que podemos obtener. Sin embargo, esta versión del cliente aún no ofrece una interfaz gráfica para su uso, lo que puede dificultar su uso. Al tratarse de un software completamente libre tampoco incluye los codecs más usuales de un escenario H.323. En consecuencia parece interesante realizar un seguimiento de este software esperando que alguna mejora substancial en el interfaz con el usuario. De cualquier forma, por el momento se incluirá en el escenario para ofrecer una gama lo más amplia posible en el acceso al servicio.

Tras este análisis llegamos a la conclusión de que la mejor opción, por el momento, es la instalación de Netmeeting en todas las máquinas que vayan a prestar este servicio de telefonía. De hecho, este software se suele instalar por defecto en las últimas distribuciones de Windows 2000, de manera que su estandarización puede resultar un punto a su favor.

- ***Gatekeeper***: este elemento proporciona diferentes servicios dentro de un sistema H.323. Entre ellos podemos destacar:

    1. Traducción de direcciones: lo cual resulta muy interesante para poder identificar a cada usuario por un alias, y no por la dirección IP de su máquina. De esta manera permitimos la movilidad de cada usuario, que puede utilizar cualquier cliente H.323 del departamento simplemente especificando su alias.
    2. Control de admisiones: permite gestionar el acceso al servicio para los diferentes usuarios. De esta manera si el administrador detecta que algún usuario está haciendo un mal uso del usuario, podrá bloquear su acceso.
    3. Control de ancho de banda: esta opción será muy interesante en un futuro cuando se implemente un sistema de gestión de calidad de servicio en la infraestructura. Por el momento no lo utilizaremos.
    4. Proporciona otros muchos servicios que no vamos a resaltar.
    5. En la actualidad se están desarrollando dentro del proyecto PISCIS otros servicios como: pasarela buzón de voz – correo electrónico, y servicios de tarificación.

El gatekeeper que utilizaremos será el de la organización www.opengatekeeper.com. La versión que utilizaremos proviene de una serie de

modificaciones sobre el software opengate_0.7alpha0. Con la inclusión de nuevo servicios pretendemos ofrecer una serie de valores añadidos a la plataforma, y de igual manera utilizarlo como realimentación para el trabajo de desarrollo que se está realizando sobre este software.

- ***Gateway***: una pasarela dentro de un entorno H.323 nos permite tener acceso a otras redes. En nuestro caso, consideramos muy interesante poder realizar llamadas a teléfonos de la RTC. Para conseguir estas comunicaciones utilizaremos la pasarela de TELDAT NUCLEOX+. De igual forma nos permitirá acceder desde un teléfono de la RTC a cualquier cliente de la plataforma.

  Este equipo dispone de cuatro líneas que podemos utilizar como teléfonos fijos o como conexión a la RTC. En nuestro caso utilizaremos un teléfono fijo para dotar de comunicación al lugar donde se encuentre situada la gateway, y utilizaremos otra línea como salida a la RTC, que se conectará a una extensión de la centralita de la universidad. De esta manera tenemos acceso directo a todos los teléfonos de la universidad dependientes de la centralita, de igual manera podemos acceder a cualquier teléfono de la RTC (si bien este servicio se ha decidido suspender).

  Es importante conocer los codecs que proporciona la pasarela, que serán los que limiten los posibles clientes que aparecerán en la plataforma. En nuestro caso el NUCLEOX+ habilita tres codecs diferentes: 1: G723 5.3Kbps. 2: G723 6.4Kbps, 3: G729A. Por desgracia el cliente Openphone no proporciona ninguno de estos tres codecs, de manera que estos clientes no podrán acceder a la pasarela, y en consecuencia a la RTC.

  Para resolver cualquier problema con la pasarela podemos consultar en el manual del administrador que se adjunta con este documento, o acceder a la web del fabricante www.teldat.es y buscar el producto NUCLEOX+.

Ante una nueva infraestructura de comunicación, como es esta que estamos creando, es muy importante elaborar un plan de numeración para permitir un acceso sencillo a todos los servicios que proporcionamos. En el apartado “Sistema de marcado” expondremos los pasos seguidos para elegir el sistema de numeración.

Descripción de equipos

Una vez descritos los elementos que formarán parte de nuestra plataforma de voz sobre IP, vamos a describir el escenario que queremos montar. Para ello lo mejor será observar el siguiente gráfico donde aparecen todos los equipos necesarios así como su distribución:

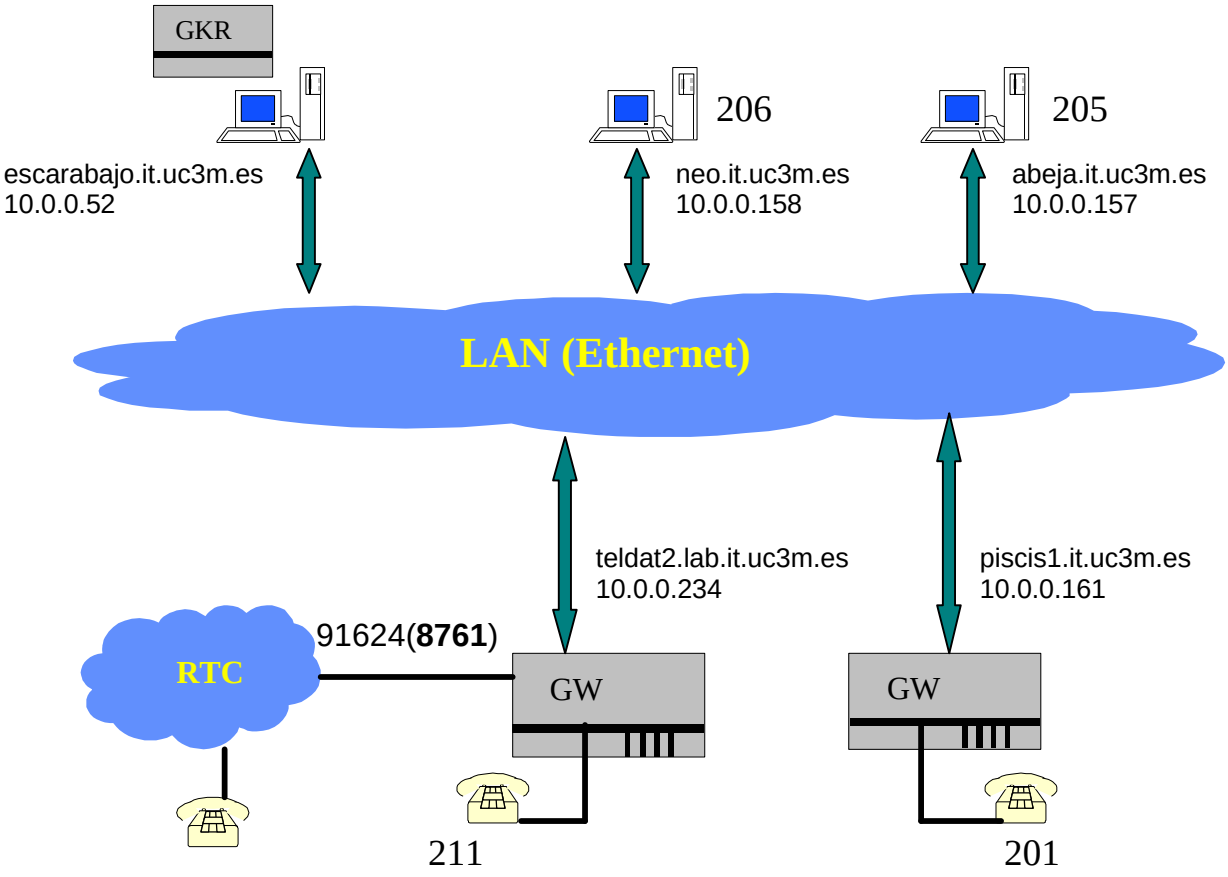


Figura III.2 – Plataforma UC3M

El gatekeeper se encuentra instalado en el equipo escarabajo.it.uc3m.es. Este software debe funcionar correctamente durante el tiempo de vida de la plataforma. Todas las modificaciones (ampliaciones de servicios) que se vayan investigando y desarrollando para el proyecto PISCIS podrán ser actualizadas aquí una vez que se compruebe su estabilidad. Escarabajo es un equipo con sistema operativo Linux SUSE 7.0, corriendo con un kernel 2.4.2. En este tenemos instalado el programa opengate_0.7alpha0. Este software, y todo el utilizado para el proyecto piscis, puede localizarse en la página web: http://www.matrix.it.uc3m.es/~piscis/software.

Existe un archivo de configuración del gatekeeper llamado opengate.ini, donde se especifica el prefijo 8, el cual utilizaremos para acceder a la centralita de la universidad a través de la gateway.

La gateway utilizada se encuentra físicamente localizada en el centro de cálculo del Departamento de Ingeniería Telemática. Se trata de un equipo NUCLEOX+ de la empresa TELDAT. Este equipo se llama teldat2.lab.it.uc3m.es. Se puede configurar mediante una comunicación telnet mediante los comandos que se indicarán en el apartado “Manual del administrador”. La pasarela dispone de un teléfono que nos permitirá comunicarnos desde la posición en que se encuentre instalada, así como un acceso a la RTC a través de la extensión 8761 de la UC3M (91-6248761). De igual forma dispone de un acceso a una red Ethernet para encaminar las llamadas a clientes de la red IP.

En la figura III.3. podemos observar un esquema con las diferentes conexiones de la pasarela.

Figura III.3 – Entorno de la pasarela

3.2.- Medida de parámetros de calidad de la plataforma VoIP

En el presente capítulo se mostrarán los resultados obtenidos a través de diferentes pruebas de calidad realizadas el 18 de Julio de 2000 sobre una serie de comunicaciones de Voz sobre IP, realizadas entre los diferentes puntos de acceso del proyecto PISCIS:

- Universidad Carlos III de Madrid
- Universidad Politécnica de Madrid
- Universidad de Sevilla

El objetivo de las presentes pruebas consiste en determinar la viabilidad de una comunicación con un nivel de calidad aceptable entre los emplazamientos anteriormente expuestos. Para ello tendremos que tener en cuenta los diferentes parámetros que influían en la calidad de una comunicación VoIP y que comentamos en el capítulo II.

Debido a la gran cantidad de software disponible para el funcionamiento como terminal de una red H.323 probaremos diferentes clientes para apreciar las diferencias que puedan existir. Estas pruebas determinarán los siguientes pasos a tomar en el desarrollo del proyecto, de manera que el escenario que montaremos finalmente incluirá los clientes que mejor se adapten a nuestro entorno y necesidades.

Según distinguimos en el capítulo II podíamos clasificar los factores que influían en la calidad de la comunicación en dos categorías:

- Inherentes al tratamiento de audio: estos factores como puedan ser eco, acoplos entre señales, ruido, etc. no representan un gran interés ya que en cualquier caso dependen de las características multimedia del equipo terminal con el que estemos trabajando. En cualquier caso realizaremos diferentes medidas de calidad donde estos parámetros estarán incluidos.

- Inherentes a la transmisión por una red IP: como comentamos anteriormente el parámetro más importante para determinar la viabilidad de una comunicación es el retardo de transmisión por la red. En consecuencia decidimos realizar unas medidas entre las diferentes universidades objeto de estudio para calcular este factor.

Dividiremos la batería de pruebas en dos escenarios diferentes. Por un lado se realizarán las pruebas de forma local en cada una de las universidades, en este caso no debería aparecer ningún problema derivado del retardo de transmisión, y nos permitirá solucionar otro tipo de problemas derivados de la configuración multimedia de los equipos. En segundo lugar realizaremos pruebas de interconexión entre las diferentes universidades donde se podrá apreciar la calidad obtenida.

3.2.1.- Modelo de Plataforma.

Cada universidad montó en su propio campus una red para la realización de pruebas internas de VoIP. Mostraremos ha continuación los equipos que componen cada una de estas plataformas, así como sus nombres y direcciones IP.

3.2.1.1.- Plataforma UC3M

El escenario de la UC3M presenta el siguiente aspecto:

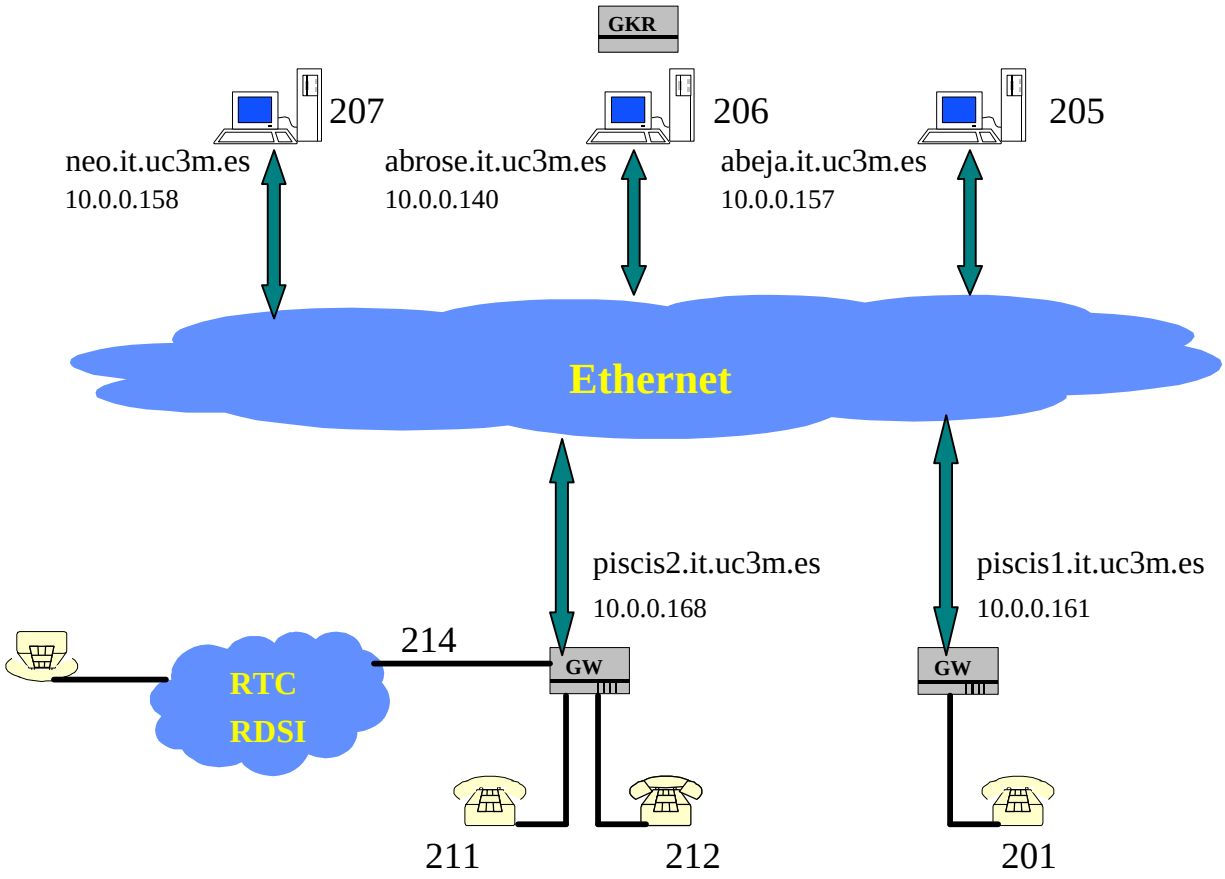


Figura III.4 – Plataforma UC3M

En la plataforma aparecen tres equipos PCs: neo, abeja y abrose. En los tres podemos encontrar los siguientes clientes VoIP: Netmeeting (versión 3.01), Openphone (versión 1.1Beta1) y Voxilla (versión 1.1Alpha1). Los dos primeros clientes operan sobre el sistema operativo Windows NT, y el último sobre Red Hat 6.2.

En la siguiente tabla se listan las direcciones IP de los diferentes elementos presentes en el escenario, así como las extensiones utilizadas para identificar los clientes en la comunicación.

MÁQUINA (alias interno)	IP	EXTENSIÓN
Abrose.it.uc3m.es (abrose)	10.0.0.140	206
Neo.it.uc3m.es (neo)	10.0.0.158	207
Piscis1.it.uc3m.es (piscis1)	10.0.0.161	201
Piscis2.it.uc3m.es (piscis2)	10.0.0.168	211 212 214 – RTC
Abeja.it.uc3m.es (abeja) Gatekeeper	10.0.0.157	

***** Los PC disponen de los clientes NetMeeting 3.01 y OpenPhone1.1beta1 sobre Windows NT, y Voxilla y OhPhone sobre Linux.

****** Sólo se indican las extensiones que tienen teléfonos conectados.

En el gateway Piscis2 se encuentra configurado el filtro 7 , que encamina el resto de número a la salida RTC. Ej: 75949, llama al 5949 por la RTC.

******* El GKR está instalado en la máquina indicada sobre Windows NT. Se ha utilizado el GKR de Opengatekeeper.

3.2.1.2.- Plataforma Universidad de Sevilla

Podemos apreciar el esquema de la plataforma de la universidad de Sevilla en la siguiente figura.

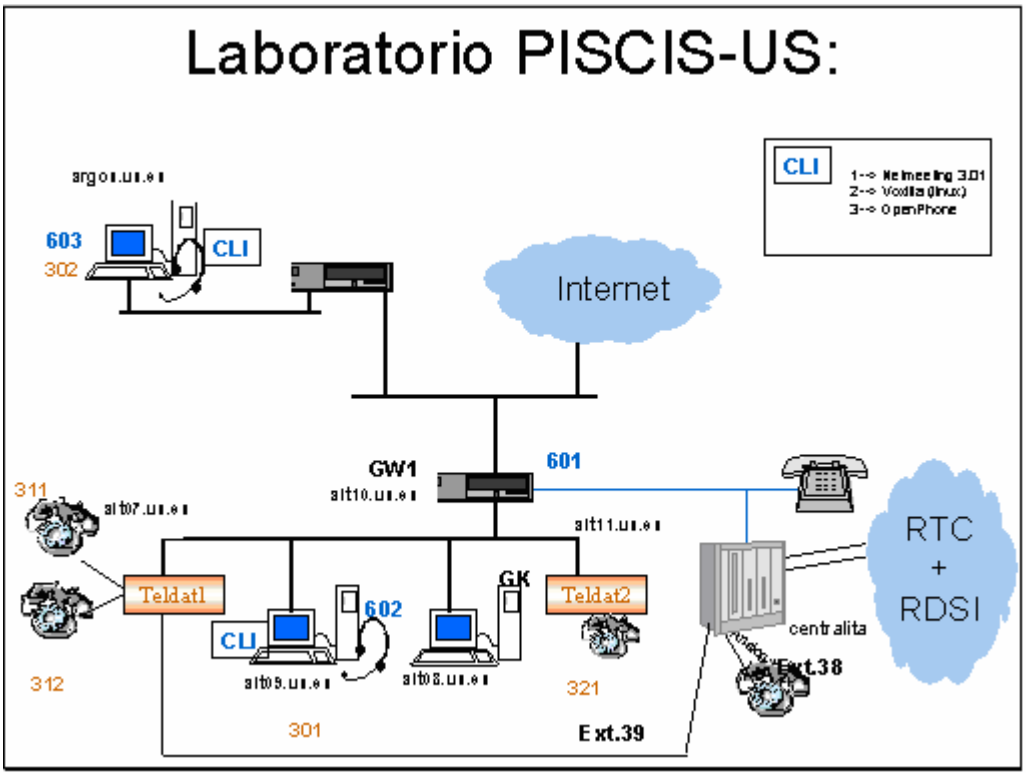


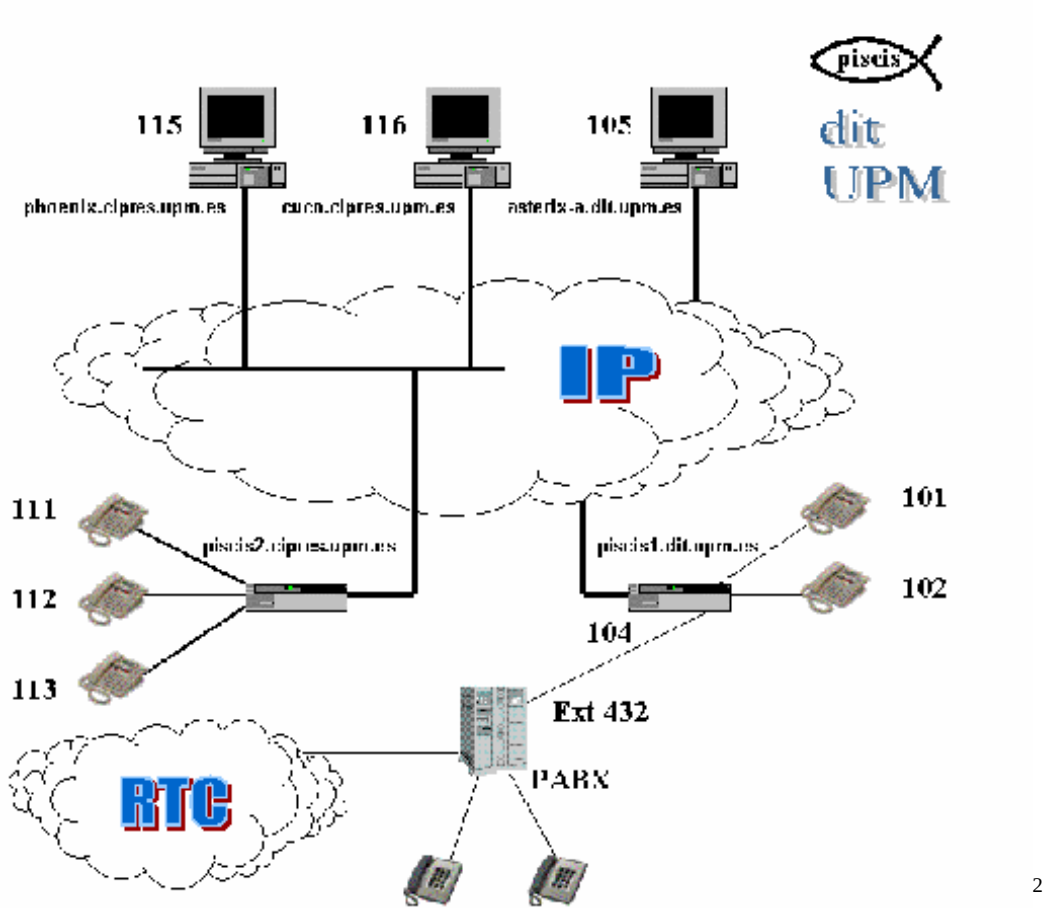
Figura III.5 – Plataforma US

En la siguiente tabla se especifican las direcciones IP de los clientes existentes así como las extensiones utilizadas en la comunicación.

MAQUINA (alias interno)	IP	EXT. TELDAT	EXT. ISDN
Argos.us.es (argos)	10.0.0.146	302	603
Ait07.us.es (teldat1)	10.0.0.168	311,312	
Ait08.us.es (piscis2)	10.0.0.169		
Ait09.us.es (piscis1)	10.0.0.170	301	602
Ait10.us.es (piscisgw)	10.0.0.171		
Ait11.us.es (teldat2)	10.0.0.172	321	
Terminal RDSI conectado a la centralita : 601			
Terminal analógico conectado a la centralita : 38			

3.2.1.3.- Plataforma Universidad Politécnica de Madrid

A continuación incluimos la arquitectura de la plataforma de la universidad politécnica de Madrid.



[2]

Figura III.6 – Plataforma UPM

Plan de numeración

	Tipo	IP address	Identificación
101	Teléfono	1 . . .48	L101
102	Teléfono	1 . . .48	L102
103	Teléfono	1 . .48	L103
104	Linea		Ext 432
105	PC	1 76	asterix-a.dit.upm.es
111	Teléfono	1 .24	L111
112	Teléfono	1 . . .24	L112
113	Teléfono	1 . .24	L113
114	Linea		Ext 7939
115	PC	1 . .23	phoenix.cipres.upm.es
116	PC	1 . .20	cuco.cipres.upm.es

[2] En el presente capítulo, por motivos de seguridad, se han eliminado o substituido las direcciones IP de los equipos presentes por direcciones privadas.

3.2.2.- Pruebas locales

3.2.2.1.- Pruebas locales UC3M

A continuación presentamos una tabla donde podemos comprobar las conexiones que han resultados satisfactorias o no entre diferentes posibles clientes H.323 y equipos TELDAT.

En concreto hemos utilizado tres clientes de H.323 bajo diferentes sistemas operativos (Linux RedHat 6.2 y Windows NT), y teléfonos conectados al router NUCLEOX+ de TELDAT y teléfonos conectados a una centralita que acceden a través de la entrada RTC del equipo TELDAT.

- **Interconexión sin gatekeeper:**

		DESTINO				
		Netmeeting	**OpenPhone**	**Voxilla**	**Teléfono IP**	**Teléfono externo**
ORIGEN	**Netmeeting**	Nivel 1	Nivel 1	Nivel 1	Nivel 1	Nivel 1
	Openphone	Nivel 1	Nivel 1	Nivel 1	Nivel 1 (1)	Nivel 1 (1)
	Voxilla	Nivel 1	Nivel 1	Nivel 1	KO(2)	KO(2)
	Teléfono IP	Nivel 1	Nivel 1 (1)	KO(2)	Nivel 1	Nivel 1
	Teléfono externo	Nivel 1	Nivel 1 (1)	KO(2)	Nivel 1	No Realizada

Notas:

1 : En el teléfono IP se escucha correctamente, pero en el OpenPhone se escucha mal o no funciona.

2 : Voxilla y los TELDATs no tienen codecs compatibles

Software utilizado:

Netmeeting 3.01 – sobre Windows NT

OpenPhone versión 1.1alpha1 – sobre Windows NT

Voxilla OhPhone versión 1.1 beta1– sobre Linux (RedHat 6.2)

Componentes:

Teléfono IP – Representa un teléfono conectado a una línea H.323 de un router TELDAT.

Teléfono externo – Representa un teléfono conectado a una centralita.

- **Interconexión con gatekeeper:**

"Opengate" para Windows NT en ordenador ajeno a todas las pruebas.

La versión utilizada del gatekeeper es opengate_0.7alpha0.50 con las bibliotecas pwlib_min_1.1pl15 y openh3231.1beta1.

		DESTINO				
		Netmeeting	**OpenPhone**	**Voxilla**	**Teléfono IP**	**Teléfono externo**
ORIGEN	**Netmeeting**	Nivel 1	Nivel 1 (1)	No Realizada	Nivel 1	KO(2)
	Openphone	Nivel 1 (1)	Nivel 1 (1)	No Realizada	Nivel 1 (1)	KO(2)
	Voxilla	No Realizada	No Realizada	No Realizada	KO(3)	KO(3)
	Teléfono IP	Nivel 1	Nivel 1 (1)	KO(3)	Nivel 1	Nivel 1
	Teléfono externo	Nivel 1 (2)	Nivel 1 (1)	KO(3)	Nivel 1	No Realizada

Notas:

1 : En el teléfono IP se escucha correctamente, pero en el OpenPhone se escucha mal o no funciona.

2 : Alto retardo de establecimiento

3 : El cliente Voxilla y los router TELDAT no presentan codecs compatibles

Software utilizado:

Netmeeting 3.01 – sobre Windows NT

OpenPhone versión 1.1alpha1 – sobre Windows NT

Voxilla OhPhone versión 1.1beta1 – sobre Linux (RedHat 6.2)

Componentes:

Teléfono IP – Representa un teléfono conectado a una línea H.323 de un TELDAT.

Teléfono externo – Representa un teléfono conectado a una centralita.

3.2.2.2.- Pruebas locales US

Con gatekeeper

		1 Netmeeting	2 Voxilla	3 Openphone	4 Teldat (directo)	5 Teldat (centralita)	6 isdn2h323
1	Netmeeting	☺	X	X	☺*	☺*	X
2	Voxilla	X	X	X	X	X	X
3	Openphone	😐	X	X	X	X	X
4	Teldat (directo)	☺*	X	X	X	X	X
5	Teldat (centralita)	☺*	X	X	X		X
6	isdn2h323	X	X	X	X	X	

sin gatekeeper

		1 Netmeeting	2 Voxilla	3 Openphone	4 Teldat (directo)	5 Teldat (centralita)	6 isdn2h323
1	Netmeeting	☺	☺	😐	☺*	☺*	☺
2	Voxilla	☺	😐	😐	X	X	☺
3	Openphone	😐	😐	😐	☺*	☺*	☺
4	Teldat (directo)	☺*	X	☺*	☺	☺	X
5	Teldat (centralita)	☺*	X	X	☺		X
6	isdn2h323	☺	☺	☺	X	X	

3.2.2.3.- Pruebas locales UPM

Pruebas Locales

	Netmeeting	Openphone	Teléfono Teldat	Teléfono Ext
Netmeeting	Nivel 1	Nivel 2	Nivel 1	Nivel 5
Openphone	Nivel 1	Nivel 3	Nivel 4	Nivel 5
Teléfono Teldat	Nivel 1	Nivel 4	Nivel 1	Nivel 5
Teléfono Ext	No test	No test	No test	No test

Nivel 1	Buena calidad de transmisión y recepción
Nivel 2	Niveles razonables de trasmisión y recepción con un pequeño retraso
Nivel 3	Hay retrasos y la voz sale metalizada
Nivel 4	Con problemas de transmisión, no es posible la comunicación
Nivel 5	No funciona

- Interconexión sin gatekeeper

3.2.3.- Pruebas externas

3.2.3.1.- Pruebas UPM-UC3M

	Netmeeting	OpenPhone	Teléfono Teldat	Teléfono Externo
Netmeeting	Nivel 1	Nivel 3	Nivel 1	Nivel 5
OpenPhone	Nivel 3	Nivel 2	Nivel 3	Nivel 5
Teléfono Teldat	Nivel 1	Nivel 3	Nivel 1	Nivel 5
Teléfono externo	No test	No test	No test	No test

Tabla III.1 – Resultado pruebas externas entre UPM y UC3M

3.2.3.2.- Pruebas US-UC3M

	Netmeeting	OpenPhone	Voxilla	Teldat-directo	Teldat-centralita
Netmeeting	Nivel 1	Nivel 2	Nivel 1	Nivel 1	Nivel 1
OpenPhone	Nivel 2	Nivel 3	Nivel 4	Nivel 4 (2)	Nivel 5
Voxilla	Nivel 1 (3)	Nivel 3 (3)	Nivel 2 (4)	No codec	No codec
Teléfono IP.	Nivel 1	No codec	No codec	Nivel 1	Nivel 1
Teléfono externo	Nivel 1 (2)	Nivel 2	No codec	Nivel 5	Nivel 1 (2)

Tabla III.2 – Resultado pruebas externas entre US y UC3M

3.2.3.3.- Pruebas US-UPM

	Netmeeting	OpenPhone	Teldat-directo	Teldat-centralita
Netmeeting	Nivel 1	Nivel 2 (2)	Nivel 1	Nivel 2
OpenPhone	Nivel 2	Nivel 2	Nivel 2	Nivel 2
Teléfono IP.	Nivel 1	No codec	Nivel 1	Nivel 5
Teléfono externo	Nivel 2	Nivel 4 (2)	Nivel 1	Nivel 5

Tabla III.3 – Resultado pruebas externas entre UPM y US

Niveles:

- Nivel 1: Buena calidad de transmisión y recepción. Sin retraso considerable.
- Nivel 2: Niveles razonables de tx y rx, con retraso apreciable.
- Nivel 3: Nivel de calidad baja, con retraso apreciable.
- Nivel 4: Se establece la comunicación pero existen grandes problemas de comprensión.
- Nivel 5: No es posible el establecimiento de la comunicación.

Notas:

1. Audio unidireccional .En US no se escucha, pero UC3M si recibe (aunque algo entrecortado).
2. Audio unidireccional. En US se escucha algo entrecortado pero UC3M no escucha..
3. Utilizando Codec G.711 u-Law 64k
4. Utilizando Codec GSM 6.10

Software utilizado:

Netmeeting 3.01 sobre Windows 98

OpenPhone versión 1.1alpha1 sobre Windows 98

Voxilla sobre Linux (RedHat 6.2)

3.2.4. Comentarios a las resultados obtenidos

En primer lugar conviene destacar que estas pruebas tuvieron lugar el 18 de Julio del año 2000, y debemos apuntar que en aquella época los conocimientos sobre una plataforma H.323 y sus componentes era muy inferior al que podamos tener en la actualidad.

En consecuencia, podemos apreciar que en muchas ocasiones aparecen problemas que impiden directamente cualquier tipo de comunicación. A partir de posteriores estudios hemos logrado descubrir la solución a muchos de los problemas que surgieron. Veamos algunos de estos casos:

- Cuando se utiliza el cliente de Microsoft Netmeeting, los codecs disponibles serán aquellos que se encuentren instalados en el sistema operativo. En una instalación normal de Windows se instalarán por defecto gran cantidad de codecs entre los que se incluyen G.711, de manera que aseguraría la compatibilidad de codecs entre clientes. Sin embargo es muy posible que alguna máquina no tuviese instalados correctamente los codecs y provocase la inviabilidad de la comunicación.

- En el caso del cliente Netmeeting, el tamaño del buffer de jitter viene determinado por la presencia del software DirectX. En el caso del sistema operativo Windows NT, este no es compatible con DirectX, de manera que presenta un buffer de jitter de aproximadamente medio segundo. Con este tamaño de buffer cualquier tipo de comunicación presenta un retardo excesivo y se considerará inviable.

- En último lugar debemos indicar que las pasarelas utilizadas (NUCLEOX+ de TELDAT) presentaba algunos problemas de estabilidad durante la realización de las pruebas, lo cual pudo llevar a impedir alguna comunicación.

A pesar de estos problemas que acabamos de comentar, podemos apreciar a partir de los resultados que la calidad obtenida en algunas ocasiones se encuentra por debajo de lo que cabía esperar. Para conocer si esto podía ser debido al retardo existente en la comunicación entre las diferentes universidades se propuso realizar medidas sobre el retardo para analizar si podría afectar excesivamente al nivel de calidad esperado.

3.2.5.- Medidas de viabilidad de la comunicación por retardo

Intentando ofrecer una mayor información sobre las posibilidades de comunicación VoIP decidimos realizar mediciones sobre el retardo de transporte existente entre los diferentes emplazamientos estudiados.

Como ya comentamos en el capítulo II el retardo debido al transporte en la red será uno de los factores más importantes en una locución VoIP. El estudio de este parámetro nos determinará si podemos esperar una calidad determinada entre dos terminales.

Para realizar estas medidas necesitamos desarrollar una aplicación que mande paquetes UDP, simulando la transmisión de audio en arquitecturas de voz sobre IP, y calcule el tiempo empleado en la transmisión y recepción del paquete. Debemos hacer notar que las aplicaciones invierten algún tiempo en construir el paquete y colocar en la tarjeta de red, si bien consideraremos este tiempo despreciable en relación al empleado para atravesar la red.

En un principio se pensó en emplear la herramienta “ping” que nos ofrecía la información que requeríamos. Sin embargo en la universidad Carlos III no se permite la salida de paquetes “ping” para evitar posibles ataques desde el exterior. Entonces nos planteamos el desarrollo de una pequeña aplicación que simulase la herramienta “ping”.

Se trata de una herramienta bastante sencilla que se divide en un servidor y un cliente. El servidor lanza un socket en la maquina destino y se asocia a un puerto determinado, entonces se queda a la espera de que alguien se conecte y mande un paquete. El cliente abre un socket en la máquina origen y manda un paquete al servidor, el cual inmediatamente responde para que la aplicación cliente pueda medir el tiempo empleado en el transporte del paquete de origen-destino más destino-origen.

El código del programa lo podemos apreciar en el apéndice correspondiente.

Debemos tener en cuenta determinados efectos provocado por el transporte UDP que influyen en el análisis de las medidas obtenidas:

- El protocolo UDP no es fiable, de manera que en alguna ocasión pueden perderse los paquetes y no alcanzar nunca el destino. Estas medidas deberán ser eliminadas ya que de otra manera falsearían los resultados que representan las condiciones reales de la red.
- Debido al uso de una red IP, cada paquete puede tomar diferentes caminos para alcanzar el destino, o alcanzar un nodo congestionado aumentado drásticamente la medida obtenida. En dichos casos y para no alterar el valor medio obtenido dichas muestras serán suprimidas. Tenemos en cuenta que durante una comunicación VoIP algunos paquetes pueden perderse, pero algunos codificadores incorporan técnicas para minimizar los efectos de estas pérdidas.
- Debido a que el retardo objeto de medida es un parámetro no determinista intentaremos realizar el mayor número posible de medidas muestrales para que están sean lo más representativas posibles.

Veamos a continuación los resultados obtenidos en las dos gráficas que aparecen a continuación.

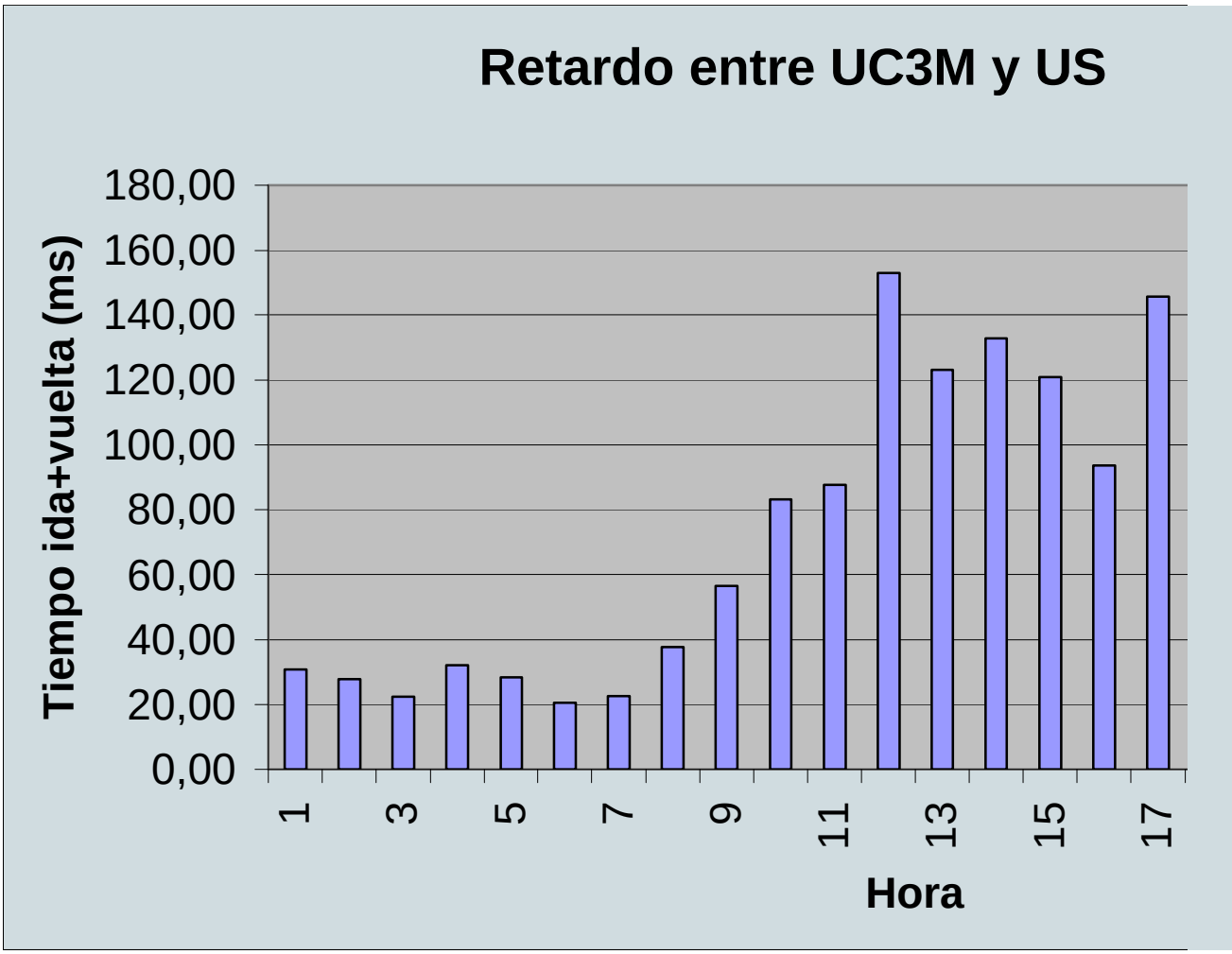


Figura III.7 - Tiempo de transmisión (+ACK) entre UC3M y US

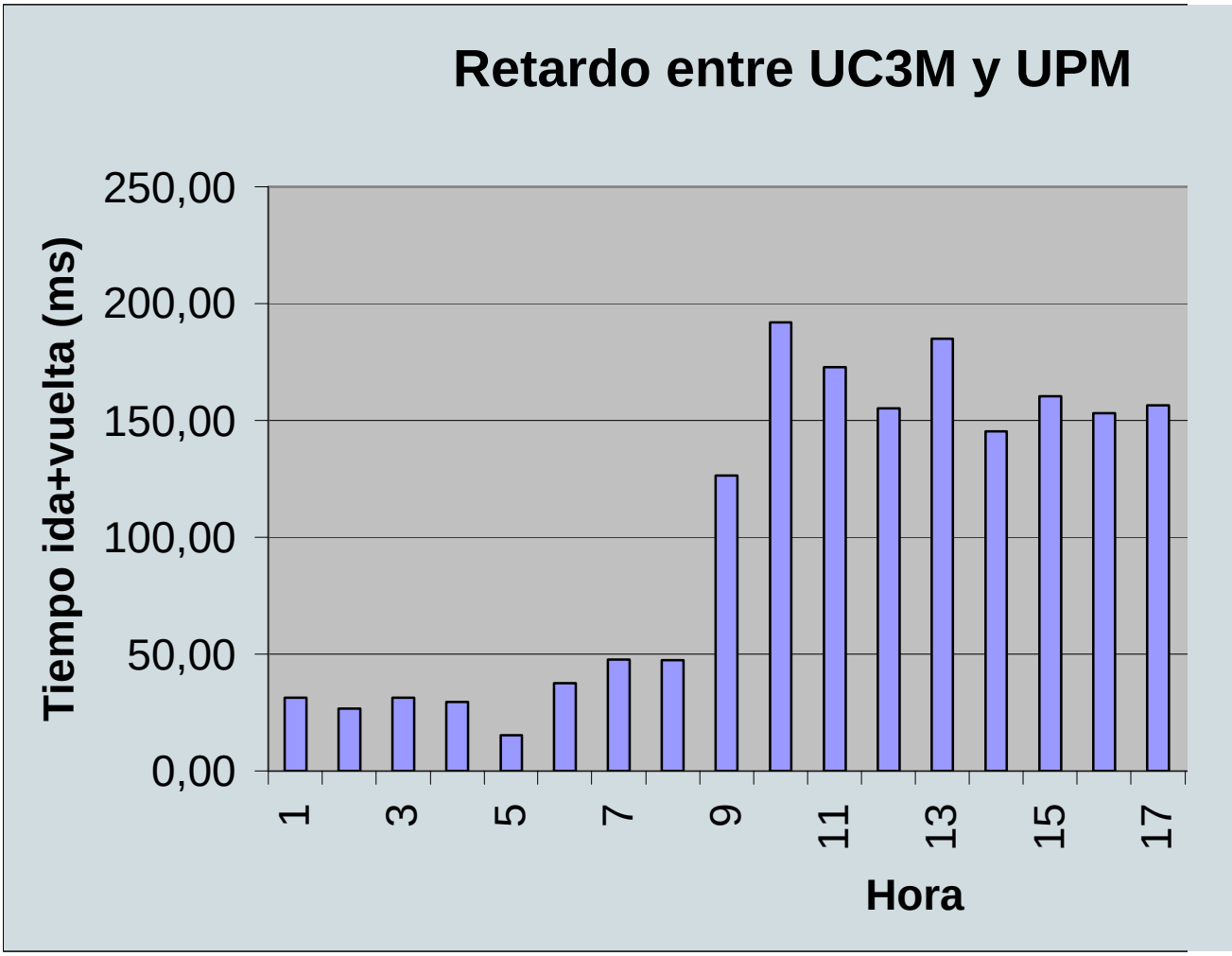


Figura III.8 - Tiempo de transmisión (+ACK) entre UC3M y UPM

Para la obtención de estas resultados se tomaron medidas durante 4 días seguidos del mes de Enero de 2001. Para obtener valores representativos se realizaron las medidas de lunes a jueves, es decir, en días laborables. De esta manera estamos tomando el caso más desfavorable, ya que el tráfico en días no laborables es muy inferior al registrado en cualquier día laboral. Por otra parte destacamos que realizamos medidas cada 20 minutos durante todo el día, para posteriormente poder calcular la media para cada periodo de una hora.

Podemos observar que en ambos casos el tiempo de transmisión nunca superaría los 100 ms. Una regla utilizada muy a menudo entre los proveedores de VoIP apunta que para conseguir alta calidad de audio este tiempo no debería superar 75mseg. En nuestro caso nos encontramos al límite de este valor, con lo cual la calidad obtenida podría no ser muy elevada.

Resulta curioso apreciar que este tiempo es mayor en el caso de la conexión con la universidad Politécnica, a pesar de encontrarse en Madrid, que con la de Sevilla. Sin embargo resulta totalmente factible, y es función de los enlaces de acceso de que dispone la Universidad Carlos III, de manera que probablemente el enlace con la Universidad de Sevilla se encuentre más descargado que el de la Politécnica.

Lo más interesante es destacar que este tiempo nos debe permitir una conexión de audio VoIP cuya calidad vendrá limitada por otros factores como el codificador o el jitter, pero nunca por el tiempo de transmisión.

Un error muy frecuente es considerar que si el tiempo de transmisión es suficientemente bajo la calidad puede ser todo lo elevada que deseemos. Sin embargo la elección del codificador fijará un límite mucho más ajustado que el de tiempo de transmisión. De esta manera la calidad que vamos a obtener nunca será mejor que la del codificador seleccionado, el cual degrada puede degradar bastante la señal.

3.3. Proyecto PISCIS

El trabajo comentado en el presente capítulo se engloba dentro de un proyecto de ámbito nacional llamado PISCIS. Para aclarar algunas de las referencias realizadas comentaremos los objetivos del proyecto.

Las siglas del proyecto PISCIS se corresponden con “Plataforma Piloto de Servicios de Comunicaciones sobre Internet de Servicios Integrados”. Se trata de un proyecto de investigación nacional que trabaja en la problemática de la transmisión de voz sobre redes de paquetes.

El principal objetivo del proyecto PISCIS es el desarrollo de una plataforma de experimentación de red que permite soportar la prestación de servicios avanzados de voz sobre redes IP con soporte de mecanismos de calidad de servicio (QoS).

El equipo de trabajo está integrado por grupos investigación de la Universidad Politécnica de Madrid, Universidad de Sevilla y Universidad Carlos III de Madrid y cuenta con la colaboración de las empresas Teldat como fabricante de equipos y SuperCable y CyC como operadores de red.

Los objetivos del proyecto sobre los que más se ha trabajado son el despliegue de una red piloto que interconecte las tres universidades utilizando técnicas de transmisión de VoIP que permitan probar la utilidad de los servicios de red desarrollados. En esta línea, se mantiene activa una plataforma entre las tres universidades.

Capítulo 4 – Integración de la plataforma en un entorno IPv6

En la actualidad el protocolo IPv6 representa un protocolo de gran importancia para empresas, consumidores y proveedores de acceso a internet. IPv6 fue diseñado para mejorar algunas características de IPv4 como pueden ser la escalabilidad, la seguridad, mayor facilidad de configuración, y la gestión de red. Estos temas son de gran importancia cuando se mide el rendimiento o competitividad de cualquier red empleada en negocios.

IPv4 puede ser modificado para implementar algunas de estas funciones, pero se supone que los beneficios obtenidos a través del despliegue de IPv6 serán mayores. Por otra parte el objetivo de IPv6 es preservar la inversión actual tanto como sea posible. Usuarios finales, ejecutivos industriales, administradores de red, ingenieros de protocolo, y muchos otros se beneficiarán conociendo la forma en que IPv6 afectará al trabajo a través de Internet.

Vistos los posibles beneficios derivados de la adaptación de nuestra plataforma a un entorno IPv6, intentaremos llevar a cabo esta migración. De esta manera nuestra plataforma conseguirá directamente las mejoras que ofrezca IPv6 y por otro lado conseguiremos una base de conocimiento sobre la migración a este protocolo.

4.1.- IPv6

El protocolo de Internet (IP) tiene sus raíces más tempranas en las redes militares de 1970 pero es en la pasada década cuando se hace imparable en el mundo de las redes. Hoy en día, IP se ha establecido por si solo como el vehículo primario para nuestro sistema global de comercio electrónico permitiendo un amplio rango de aplicaciones cliente servidor.

La Internet Engineering Task Force (IETF) ha producido un conjunto de especificaciones (RFC 1752, 1883, 1886, 1971, 1973, etc.) que definen la siguiente generación de protocolo IP conocido como "IPNG" o IPv6.

IPv6, la siguiente generación de protocolo de Internet, fue aprobado por el Internet Engineering Steering Group el 17 de Noviembre de 1994 como una propuesta de estándar. Desde ese momento un gran número de organizaciones de usuarios finales, grupos de estándares y vendedores de redes han estado trabajando juntos en la especificación y pruebas de implementaciones de IPv6.

Un gran número de grupos de trabajos de IETF han definido el proyecto de IPv6 incluyendo especificaciones de protocolo, arquitectura de direcciones, seguridad, mecanismos de transición, DNS (Domain Name Server) y ICMP (Internet Control Message Protocol)**.**

IPv6 es un término sencillo pero que abarca un gran campo para los propietarios de redes y proveedores de servicios, los productos de IP ya están en el mercado y así continuarán hasta la próxima década.

Con IPv4 al contrario que con el sistema telefónico, que utiliza un código de país y de área, la numeración no era jerárquica. Los bloques de números se asignaban a organizaciones de manera muy ineficiente, desperdiciándose gran parte del espacio de números.

El resultado es que el espacio de números se ha agotado. Además como los números no se asignan jerárquicamente las tablas de enrutamiento crecen muy rápidamente. No se espera una reducción de la expansión de Internet. Además, aparecen nuevos desafíos:

- La comunicación de las nuevas generaciones de computadoras personales móviles y los asistentes personales digitales.

- La anticipación de la demanda de audio y vídeo en tiempo real, que pondrá la actual tecnología en sus límites.

El mundo del comercio se ha trasladado a Internet y ha quedado claro que ya es hora de crear una infraestructura segura de red.

El desarrollo de IPv6, se ha visto estimulado por la urgente necesidad de resolver los problemas de direcciones de Internet, enrutamiento, rendimiento, seguridad y congestión.

Así como también permite la definición flexible y jerárquica de una arquitectura de enrutamiento global con varios niveles.

4.1.1.- Características de IPv6

Las características de IPv6 son las siguientes:

- Dispone de direcciones de 128 bits, 16 octetos, que se pueden estructurar jerárquicamente para simplificar la delegación de direcciones y el enrutamiento; garantiza una única dirección para cada dispositivo de red..
- Simplifica la cabecera principal IP, pero define muchas cabeceras de extensión opcionales. De esta forma se pueden incorporar las nuevas funciones de intercomunicación cuando lo necesiten.

- Dispone de autenticación, integridad de datos y confidencialidad en el nivel de IP.
- IPv6 es principalmente relevante para enrutadores de "backbone" y no para aplicaciones de usuario final. Muchas de las bondades de IPv6 traen beneficios directos a las aplicaciones de usuario final en los niveles de grupo de trabajo, pero también se tienen nuevas propiedades de encriptación y servicios de autenticación que permite una asignación eficiente de direcciones IP sin retardos y costos asociados con la administración manual de direcciones.

- Introduce flujos, que se pueden utilizar para disponer de nuevos tipos de requisitos de transmisión, como el vídeo en tiempo real.

- Facilita el encapsulado de otros protocolos y proporciona un mecanismo de control de congestión cuando transporta protocolos extraños.

- Proporciona nuevos métodos de autoconfiguración automática de direcciones e incorpora una comprobación de que las direcciones son únicas.

- Mejora el descubrimiento del encaminador y la detección de encaminadores fuera de servicio o vecinos inalcanzables por el enlace.
- Calidad de servicio integrada (QoS).
- Autoconfiguración, computación móvil y un agregado más eficiente de rutas de red a nivel de "backbone" global.

Pueden servir en conjunto con ATM para efectuar muy diferentes y complementarios roles en redes corporativas. IPv6, al igual que su predecesor, provee servicios de capa de red sobre la mayoría de los tipos de enlaces incluyendo ATM, Ethernet, Token ring, ISDN, Frame Relay y T1.

4.1.2.- Calidad de servicio en IPv6

El formato de paquetes en IPv6 contiene un nuevo campo de identificación de flujo de tráfico de 24 bits que tendrá un gran valor para vendedores que implementan funciones de red de calidad de servicio.

Las etiquetas de flujo en IPv6 pueden ser usadas para identificar en la red un conjunto de paquetes que necesitan un manejo especial durante y después de una falla. El enrutamiento basado en flujo podría mejorar algunas de las características determinísticas asociadas con la tecnología de conmutación orientada a conexión y los circuitos virtuales telefónicos.

El flujo es una secuencia de paquetes desde un origen a un destino que necesita de cierto tratamiento especial, por lo tanto la etiqueta de flujo se utiliza para identificar un flujo de datos que tiene un mecanismo de manejo especial (por ejemplo, reserva de ancho de banda).

4.1.3.- Seguridad de IPv6

IPv6 provee propiedades de seguridad de datos basadas en sus extensiones de cabeceras flexibles.

La extensión de cabecera de autenticación para IPv6 asegura que un paquete está actualmente viniendo desde el "host" indicado en su dirección fuente.

Las cabeceras de autenticación en IPv6 no proveen privacidad o confidencialidad de datos, estas funciones son realizadas con otra extensión de cabecera estándar que da encriptación de principio a fin en la capa de red.

Las cabeceras de encriptación tienen campos que llevan las claves de encriptación y otras informaciones, permitiendo así la encriptación interoperante de paquetes IP.

Las cabeceras de seguridad IPv6 pueden ser usadas directamente entre los "hosts" o en conjunto con dispositivos de seguridad que añade un nivel adicional de seguridad con sus propios método de encriptación y paquetes de señalización.

4.1.4.- Ventajas de IPv6

- **Simplificación del formato del encabezado**

  Algunos campos del encabezado de IPv4 habían disminuido o eran opcionales para reducir los costos de procesamiento del manejo de paquetes.

  Con IPv6 se mantienen bajos los costos del ancho de banda a pesar de que se cuadriplicaron las direcciones, sin embargo el encabezado se redujo a la mitad.

- **Mejora del soporte para las opciones**

  Los cambios en la forma en que las opciones del encabezado IP son codificados permiten mayor eficiencia en el envío, menos limitaciones en la longitud de las opciones y mayor flexibilidad para incorporar nuevas opciones en el futuro

- **Capacidades en la Calidad del Servicio**

Una nueva capacidad se incorporó para habilitar el marcaje de paquetes que pertenecen a un "flujo" de tráfico particular, para el cual el remitente requiere un manejo especial, tal como un servicio en tiempo real o un servicio de alta calidad.

- **Autenticación y Capacidad de Privacidad**

  IP incluye la definición de extensiones las cuales suministran soporte para autenticación, integridad de los datos y confidencialidad. Esto se incluye como un elemento básico de IPNG y se incorporará en todas las implementaciones.

Existe un grupo de trabajo denominado Internet Engineering Task Force (IEFT) que está trabajando en conjunto con otras organizaciones en el desarrollo de algoritmos que "mapeen" las direcciones tanto en IPv6 como en otros ambientes. Un ejemplo de ello son los algoritmos de "mapping" que ya existen o están bajo desarrollo orientados a varios tipos de direccionamiento incluyendo algoritmos de "mapping" para direcciones en redes Novell IPX, algunos tipos de Puntos de Acceso de Servicios de Redes OSI (NSAP), las direcciones E164 y direcciones SNA. Tales "mapping" proveerán un mapa ("map") uno - a - uno entre enrutadores y subredes.

Al disminuir el campo en el encabezado y dándole una longitud fija incrementan la eficiencia de IPv6.

4.1.5.- Futuro de IPv6

A pesar de su fiabilidad, IPv4 tiene, entre otras deficiencias, un campo de direcciones muy reducido, problemas de rendimiento y algunos agujeros de seguridad, y todas ellas han de ser resueltas para poder dar cauce a las tecnologías Internet e IP,

especialmente teniendo en cuenta la popularidad creciente de la red y de su utilización como medio de transmisión de información multimedia.

IPv6, la nueva generación de Internet Protocol, es la respuesta, además de ampliar la capacidad del campo de direcciones, IPv6 está diseñada para superar otras limitaciones de la versión actual, como la calidad de servicio y la configuración de enrutadores y hosts. Pero la adopción de IPv6 requiere ciertos cambios en las aplicaciones, los protocolos de encaminamiento y los servidores de direcciones, según afirman los desarrolladores y fabricantes que están actualmente envueltos en la red de prueba de IPv6 6bone. Pero lo que realmente puede dificultar y entorpecer la migración a IPv6 son las aplicaciones, pues no podrán trabajar con el nuevo protocolo si antes no se adaptan a él.

Algunos observadores, incluso, cuestionan la necesidad de adoptar IPv6 si los inconvenientes propios de IPv4 pueden ser resueltos de otro modo.

Entre todas las ventajas prometidas por el nuevo protocolo, las más importantes son las mejoras conseguidas en las direcciones; hasta el punto de que justifican por sí solas el propio desarrollo de la versión 6. Según los desarrolladores, el campo de direcciones de 32 bits de IPv4 se ha quedado muy pequeño para satisfacer tanto el creciente uso de Internet como la expansión constante de los requerimientos de ancho de banda y de potencia de procesamiento.

IPv6 proporciona un campo de direcciones de 128 bits que incrementa exponencialmente el número de dispositivos que puede soportar el protocolo en comparación con IPv4. Algunos observadores predicen incluso que las direcciones de la versión 4 se agotarán dentro de cinco u ocho años. Esa es una de las razones por las que Internet Engineering Task Force (IETF) publicó un RFC (request for comment) en 1993 donde se recogen sugerencias para afrontar la migración.

Los métodos de migración recomendados por el IETF son la utilización de dos pilas de protocolos y encapsulamiento. El primero se refiere a la disposición de nodos IP capaces de soporta tanto protocolos IPv6 como IPv4. El enfoque de encapsulamiento (efecto túnel) se basa en transmitir paquetes IPv6 sobre las infraestructuras IPv4 actuales. Los fabricantes aseguran que, en la práctica, ambas técnicas deberían

minimizar cualquier problema de migración. Existen herramientas que permiten acometer planes de migración en distintas fases. No es necesario migrar a un tiempo todos los enrutadores a IPv6. Se pueden tener islas de conectividad IPv6 conectadas a mecanismos de encapsulamiento. Pero los usuarios no confían del todo en estos mensajes de calma porque, a pesar de las herramientas de migración incorporadas, es casi seguro que la transición a IPv6 dará más de un problema. La mayoría aflorarán cuando los usuarios tengan que modificar sus aplicaciones para trabajar en el nuevo entorno de red. Por ejemplo, el campo de interfaz de usuario en una aplicación escrita para IPv4 habrá de ser ampliado para que pueda tratar las mayores direcciones de IPv6. Además, los usuarios o desarrolladores tendrán que cambiar el modo en que las aplicaciones pasan las direcciones a la interfaz WinSock a nivel de red. WinSock es una interfaz de programación de aplicaciones de los sistemas Windows 95 y Windows NT que une las aplicaciones a las pilas de protocolos TCP/IP. Algunos fabricantes recomiendan que los usuarios hagan inventario de sus aplicaciones antes de desplegar IPv6, a fin de minimizar los imprevistos. Una tarea crítico que los usuarios deben de acometer cuanto antes.

Los usuarios que quieran evitarse preocupaciones y deseen seguir usando IPv4, pueden utilizar NAT (Network Address Transalation) para ampliar el número de direcciones disponibles. Los servidores NAT, en el límite entre las intranets privadas e Internet, permiten a los usuarios aumentar el uso de las direcciones estableciendo una distinción entre direcciones de red privada y direcciones Internet. Para ahorrar direcciones Internet, NAT las asigna sólo a aquellos usuarios Internet activos, de modo que cuando éstos desconectan de la Red la dirección regresa a un pool compartido. Así, las organizaciones pueden satisfacer sus necesidades de conexión a Internet con un número mucho más reducido de direcciones. Esta solución está indicada especialmente para aquellas empresas que consideren que la mejora ofrecida por IPv6 no compensa el esfuerzo que supone adoptarlo.

Eso no quiere decir, ni mucho menos, que los usuarios no necesiten ni deban implementar la nueva versión de IP. De hecho, NAT es una solución no demasiado escalable una vez que la red comienza a crecer y volverse más jerárquica. Con todo, los observadores y consultores coinciden en que IPv6 no generará una migración en masa, al menos hasta dentro de algunos años.

4.2.- Migración IPv4-IPv6

En los apartados anteriores hemos visto las nuevas características que incorpora IPv6 respecto a la anterior versión IPv4. Ahora estudiaremos los cambios necesarios para portar una aplicación a la versión 6 del protocolo IP. En el presente capítulo nos gustaría describir los cambios que se deben introducir a nivel de programación. Para ello el entorno de trabajo será un sistema operativo Linux con el kernel 2.4.2. Para poder trabajar con IPv6 es necesario recompilar el kernel activando la opción de soporte IPv6.

Una vez identificado el programa que se desea transformar, así como la localización de la biblioteca de comunicaciones deberemos tener en cuenta las siguientes características de la nueva versión:

- Mientras que las direcciones IPv4 tienen una longitud de 32 bits, las direcciones IPv6 ocupan 128 bits. En consecuencia será necesario modificar la longitud de las variables donde se almacenasen direcciones IP.

  Debemos tener en cuenta que las direcciones IP en su notación estándar viajarán a lo largo de todo el programa como cadenas de caracteres, de manera que tendremos que analizar todos los lugares por donde se intercambien estas direcciones.

- En el protocolo IPv6 se introducen nuevos campos como traffic class y flow label que pertenecen a la nueva cabecera IPv6. En caso de que la aplicación no acceda a estos parámetros simplemente deberemos encargarnos de inicializarlos correctamente.

- La mayoría de las funciones referentes al interfaz de comunicaciones han sido reemplazadas por otras que permiten el soporte IPv6, de manera que será necesario buscar todas estas funciones y cambiarlas por sus adaptaciones.

- Resultaría muy interesante que las aplicaciones no dependiesen del protocolo sobre el que deben trabajar. Si bien esto requeriría una doble fase de programación que distinguiese la dirección IP escogida para detectar si hace uso de IPv4 o IPv6. La programación sería más sencilla si se decidiese conseguir soporte único sobre IPv6.

4.2.1.- El API de comunicaciones de Linux

El sistema operativo Linux nos ofrece un conjunto de funciones que facilitan el trabajo con los elementos hardware encargados del transporte en red. Estas funciones forman lo que llamaremos el API del interfaz de socket. Podemos dividir la composición de este API en los siguientes grupos:

- Funciones del núcleo del socket
- Estructuras de datos
- Funciones de traducción de nombres a direcciones
- Funciones de conversión de direcciones

Las funciones núcleo del socket son aquellas encargadas del establecimiento y liberación de las conexiones, así como del transporte de paquetes UDP. Estas funciones se diseñaron para ser independientes del protocolo de transporte.

Se define una estructura de datos de direcciones específicas del protocolo para cada protocolo que las funciones del socket soportan.

Las aplicaciones deben mapear los punteros a estas estructuras específicas en punteros a la estructura de direcciones genérica "sockaddr" cuando emplean las funciones del socket.

Estas funciones no se necesitan cambiar para IPv6, pero una nueva estructura de datos de direcciones necesita ser introducida para IPv6.

La estructura "sockaddr_in" es la estructura de datos específica de IPv4. Puesto que no es suficiente para contener las direcciones de IPv6, una nueva estructura de datos necesita ser introducida para dar soporte a IPv6.

Las funciones de traducción de direcciones en IPv4 son gethostbyname() y gethostbyaddr(). Éstas funciones se dejan como están y se definen nuevas funciones que den soporte tanto a IPv4 como a IPv6.

Las funciones de conversión de direcciones son inet_ntoa() e inet_addr(), usadas para convertir direcciones IPv4 entre forma binaria e imprimible.

De nuevo se han diseñado funciones equivalentes que den soporte tanto a IPv4 como a IPv6.

4.2.2.- Adaptaciones realizadas

Veamos a continuación los puntos más destacados que han tenido que ser adaptados en el interfaz de socket para permitir la utilización del protocolo IPv6:

Familia de protocolo IPv6 y familia de direcciones

Con la aparición de la nueva versión del protocolo es necesario definir una nueva familia de direcciones, AF_INET6, cuya definición podemos encontrar en <sys/socket.h>. En esta definición se distingue entre la nueva estructura de datos de direcciones sockaddr_in6 y la antigua utilizada en IPv4, sockaddr_in.

De igual forma que con la familia de direcciones, se define un nombre de familia de protocolos, PF_INET6, localizado en <sys/socket.h>, que por definición tendrá el mismo valor que la familia de direcciones.

```
#define PF_INET6        AF_INET6
```

Este valor se utiliza en la llamada a la función socket, como primer argumento, he indica que el nuevo socket creado pertenecerá al protocolo IPv6.

Estructura de direcciones IPv6

Debido a la variación del tamaño de dirección en la nueva versión del protocolo será necesario redefinir la estructura donde se almacenaba esta información. Esta información se encuentra disponible en <netinet/in.h> y nos facilita la siguiente estructura:

```
struct in6_addr {
 uint8_t  s6_addr[16];      /* IPv6 address */
  };
```

Podemos apreciar como la dirección IPv6 se almacena en un array de 16 elementos de 8 bits.

Interfaz del socket

En la interfaz del socket podemos localizar una estructura de datos que incorpora información especifica de cada protocolo. En esta estructura se almacenará información referente a cada comunicación, que será mapeada directamente sobre diferentes campos del protocolo. Está estructura es llamada “sockaddr”.

Para IPv4, la estructura de datos específica es sockaddr_in, empleada como hemos indicado para pasar direcciones entre las aplicaciones y el sistema por medio de las funciones del socket. En la nueva versión del protocolo, IPv6, emplearemos la estructura sockaddr_in6, la cual aparece definida en la biblioteca <netinet/in.h>.

```
struct sockaddr_in6 {
```

```
    sa_family_t     sin6_family;    /* AF_INET6 */
    in_port_t       sin6_port;      /* transport layer port # */
    uint32_t        sin6_flowinfo;  /* IPv6 traffic class & flow info */
    struct in6_addr sin6_addr;      /* IPv6 address */
    uint32_t        sin6_scope_id;  /* set of interfaces for a scope */
};
```

En el campo sin6_family se indica la familia de protocolos que en nuestro caso deberá ser AF_INET6, identificando la dirección como una de clase IPv6.

El campo sin6_port contiene el número de puerto UDP o TCP tal como sucedía en IPv4.

El campo sin6_flowinfo contiene la clase de tráfico y el flowlabel, especifica de IPv6.

El campo sin6_addr contiene la dirección IPv6.

Función socket

La función socket es la encargado de generar un descriptor que identificará nuestro canal de comunicación. Esta función debe recibir los argumentos apropiados para identificar el tipo de protocolo, así como las características deseadas.

Así por ejemplo, si pretendemos crear un socket para el protocolo IPv4 que funcione con capa de transporte UDP utilizaríamos:

```
s = socket(PF_INET, SOCK_DGRAM, 0);
```

El mismo ejemplo para el protocolo IPv6 sería muy similar:

```
s = socket(PF_INET6, SOCK_DGRAM, 0);
```

A través de este descriptor, y utilizando las estructuras de direcciones sockaddr_in6 podremos enviar y recibir paquetes.

Opciones del socket

Como en el caso del protocolo IPv4, es posible seleccionar algunas opciones referentes al tráfico unicast y multicast. Para ello se utilizan las funciones getsockopt() y setsockopt(). Todas comienzan por IPv6, para distinguirlas claramente de las de IPv4.

Para disponer de ellas será necesario incluir la cabecera <netinet/in.h>.

Las opciones más importantes son:

IPV6_UNICAST_HOPS : empleada para limitar el número de hops .
IPV6_MULTICAST_IF : habilita la interfaz para enviar paquetes multicast.
IPV6_MULTICAST_HOPS : limita el número de hops en multicast.
IPV6_JOIN_GROUP : permite unirse a un grupo multicast.
IPV6_LEAVE_GROUP : abandona un grupo multicast.

Conversión de direcciones de forma binaria a imprimible

Durante la programación resulta muy interesante disponer de unas funciones que permitan la transformación de direcciones entre un formato texto, que representa una dirección en el formato típico, y un formato binario que es el que almacenamos en las estructuras de direcciones anteriormente citadas (sockaddr_in6).

Antiguamente, para el protocolo IPv4 se utilizaban las funciones inet_ntoa e inet aton. Con la llegada del protocolo IPv6 se crearon dos nuevas funciones compatibles para ambos protocolos:

```
#include <sys/socket.h>
#include <arpa/inet.h>

int inet_pton(int af, const char *src, void *dst);

const char *inet_ntop(int af, const void *src,
        char *dst, socklen_t size);
```

La primera de ellas convierte una dirección en forma de texto a una en forma binaria. El argumento af especifica la familia de direcciones. src y dst son punteros a las direcciones origen y destino (argumentos de entrada y de salida de la función).

La segunda función realiza el procedimiento inverso.

4.3.- Migración de clientes SIP

Una vez realizado el estudio previo para descubrir los cambios que deberíamos introducir para lograr que la aplicación soportase el protocolo IP en su versión 6, buscamos alguna aplicación adecuada que nos permitiese introducir estos cambios.

En un principio se planteo la posibilidad de migrar la aplicación OhPhone, uno de los clientes H.323 utilizados en la plataforma VoIP. Para comenzar el trabajo se descargo el código fuente de dicha aplicación desde la siguiente dirección de Internet:

OhPhone_1.1.4	**http://www.openh323.org/code.html**

En un principio podemos esperar que las funciones referentes al interfaz de comunicaciones se encuentren en algún archivo separado del programa principal, o incluso, en una biblioteca externa. Para realizar la búsqueda de cualquier indicio la forma más adecuada es buscar la aparición de la función socket a lo largo del código fuente haciendo uso de la herramienta “grep” bajo linux.

En nuestro caso encontramos que todas las funciones referentes a comunicaciones se encontraban en otra biblioteca externa denominada PWLIB utilizada por otras muchas aplicaciones. PWLib es una biblioteca de tamaño considerable que permite crear aplicaciones ejecutables tanto bajo Windows como Unix. Esta biblioteca proporciona funciones de entrada/salida, multihilos, protocolos Internet, etc.

Modificar una biblioteca como PWLib se salía de las aspiraciones de nuestro proyecto de manera que se decidió realizar otra búsqueda sobre diferentes aplicaciones para que la migración de dicha aplicación no fuese extremadamente compleja y no s permitiese extraer los conocimientos para futuras conversiones.

Todas las aplicaciones del proyecto OpenH323 utilizan la biblioteca PWLib como soporte de manera que decidimos buscar otra aplicación sobre la cual pudiésemos trabajar más fácilmente aumentando así el rendimiento de nuestro trabajo.

Para ello se decidió realizar un estudio sobre el cliente de comunicaciones del protocolo SIP. Al tratarse de un protocolo algo más sencillo que H.323 cabía esperar que la biblioteca de comunicaciones fuese igualmente algo menos compleja. El software lo pudimos descargar desde:

Sipc v1.5	**http://www.cs.columbia.edu/IRT/software/**

Como en el caso anterior realizamos una búsqueda de las funciones socket utilizadas con la intención de localizar la posible biblioteca de comunicaciones. En este caso encontramos que todas estas funciones se encontraban localizadas en el fichero *sipc.src/tcludp/src/udp_tcl.c.* Una vez localizado este fichero realizamos las modificaciones necesarias que hemos comentado en el apartado anterior y que resumimos a continuación:

- En primer lugar tenemos que tener en cuenta las nuevas estructuras utilizadas. Deberemos cambiar las definiciones *struct sockaddr_in* por *struct sockaddr_in6*. De esta forma las nuevas estructuras podrán almacenar las direcciones IPv6 así como otros parámetros nuevos que aparecen en la nueva versión.
- Debido a este cambio de las estructura todas las llamadas a las funciones *sendto, recvfrom, connect, bind* deberán modificar sus parámetros para adaptarse a las nuevas estructuras.
- Deberemos tener en cuenta la inicialización de los nuevos parámetros de esta estructura como *flowinfo*.
- El siguiente paso sería modificar en todas las llamadas a sockets la familia de protocolos desde AF_INET a AF_INET6.
- Finalmente queda reemplazar el uso de antiguas funciones que no permitían soporte IPv6 por otras nuevas que aportan esta nueva funcionalidad.

Comentemos cuales son estas funciones y por cuales fueron reemplazadas:

Antigua Función	Nueva Función	Descripción
Inet_ntoa	Inet_ntop	Convierte la dirección del host dada en formato binario a una cadena con la notación estándar de IPv4/IPv6.
Inet_aton	Inet_pton	Convierte la dirección del host desde la notación estándar de IPv4/IPv6 a datos binarios y los almacena en la estructura especificada.
Gethostbyname	Getaddrinfo	Permite obtener la dirección IP a partir del nombre de la maquina. Es decir, realiza una consulta al DNS.

Una vez realizadas estas modificaciones comenzó el proceso de depuración para solucionar los posibles fallos que apareciesen en el software generado. Para ello además de utilizar un fichero con logs referentes a las operaciones de red utilizadas nos ayudamos de herramientas como *Ethereal* que permite visualizar todos los paquetes que circulan por la red. Y también nos apoyamos en la utilidad *strace* que permite visualizar con mayor profundidad las llamadas realizadas por el sistema operativo.

A partir de la depuración pudimos localizar diferentes problemas y solucionarlos. Adicionalmente aparecieron otra serie de problemas características de nuestra aplicación en concreto, si bien resultan de gran interés ya que pueden aparecer en otras muchas aplicaciones:

- En primer lugar aparece un problema derivado de la nomenclatura de direcciones IPv4 e IPv6. Como todos sabemos las direcciones IPv4 constan de cuatro cifras separadas por un punto, y en algunas ocasiones aparece indicado el puerto al final separado con dos puntos. En el caso de IPv6 todas las cifras que componen la dirección aparecen divididas por dos puntos.

En nuestra aplicación, sipc, para evitar problemas con los puertos cortaba la dirección IPv4 si encontraba el carácter ‘:’. De esta manera nos imposibilita a introducir direcciones IPv6. Para superar este obstáculo se decidió introducir las direcciones IPv6 separadas con el carácter ‘.’ en lugar de ‘:’, e internamente, el programa realizaba la conversión entre estos caracteres.

- Por otro lado en un principio no se recibía ningún mensaje del receptor, de manera que probablemente no pasábamos correctamente la dirección del origen al cliente destino. El receptor no puede obtener esa dirección a partir de cualquier mensaje recibido, ya que dicho mensaje puede haber pasado por algún servidor proxy, de manera que dicha dirección se incluye dentro del paquete SIP.

  Comprobamos que dicha dirección la obtenía a través de una llamada al sistema con */sbin/ifconfig*. Este comando devuelve todas las direcciones configuradas en la maquina para todos los interfaces. Nuestra aplicación buscaba la dirección localizando la cadena “*inet addr*”. Sin embargo la nueva dirección IPv6 viene identificada por la cadena “*inet addr6*”. Fue necesario introducir unos cambios sobre parte de software escrito en TCL para salvar este pequeño inconveniente.

Nos gustaría resaltar que si bien parte de los cambios a introducir para migrar una aplicación desde IPv4 a IPv6 son bastante mecánicas y se encuentran muy bien documentados, cada aplicación se encuentra programada de diferentes formas, y siempre se encuentran diferentes peculiaridades que requieren de un estudio individual para migrar adecuadamente cualquier aplicación.

Demostración del correcto funcionamiento

A continuación mostramos la figura IV-1 donde podemos apreciar el entorno de trabajo. En el aparecen dos clientes sipc corriendo en diferentes máquinas corriendo la aplicación chat sobre ellos.

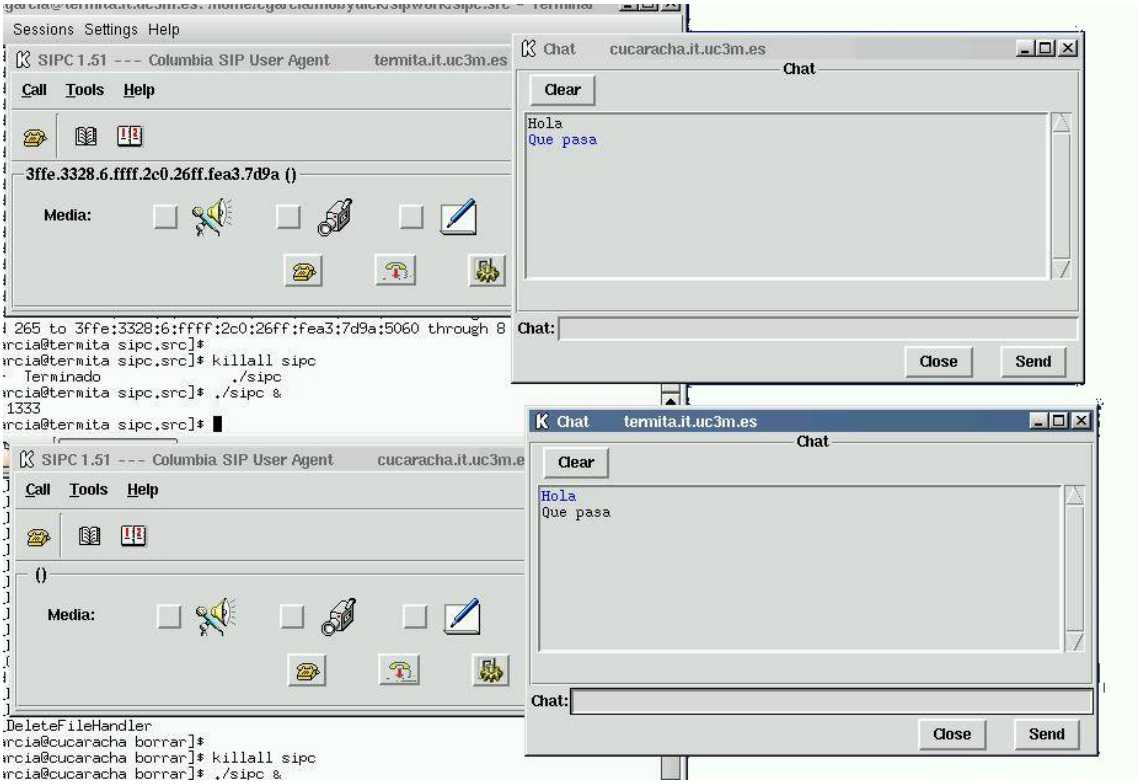


Figura IV.1 – Entorno de trabajo

Según comentamos anteriormente utilizamos la herramienta Ethereal para comprobar el intercambio de paquetes entre ambos clientes. Para ello configuramos Ethereal para filtrar solo los paquetes IPv6 y que contengan la dirección IPv6 de alguna de las máquinas implicadas. En la figura IV-2 podemos observar el inicio de una sesión SIP entre ambas máquinas, así como el intercambio de un paquete de datos de la aplicación chat.

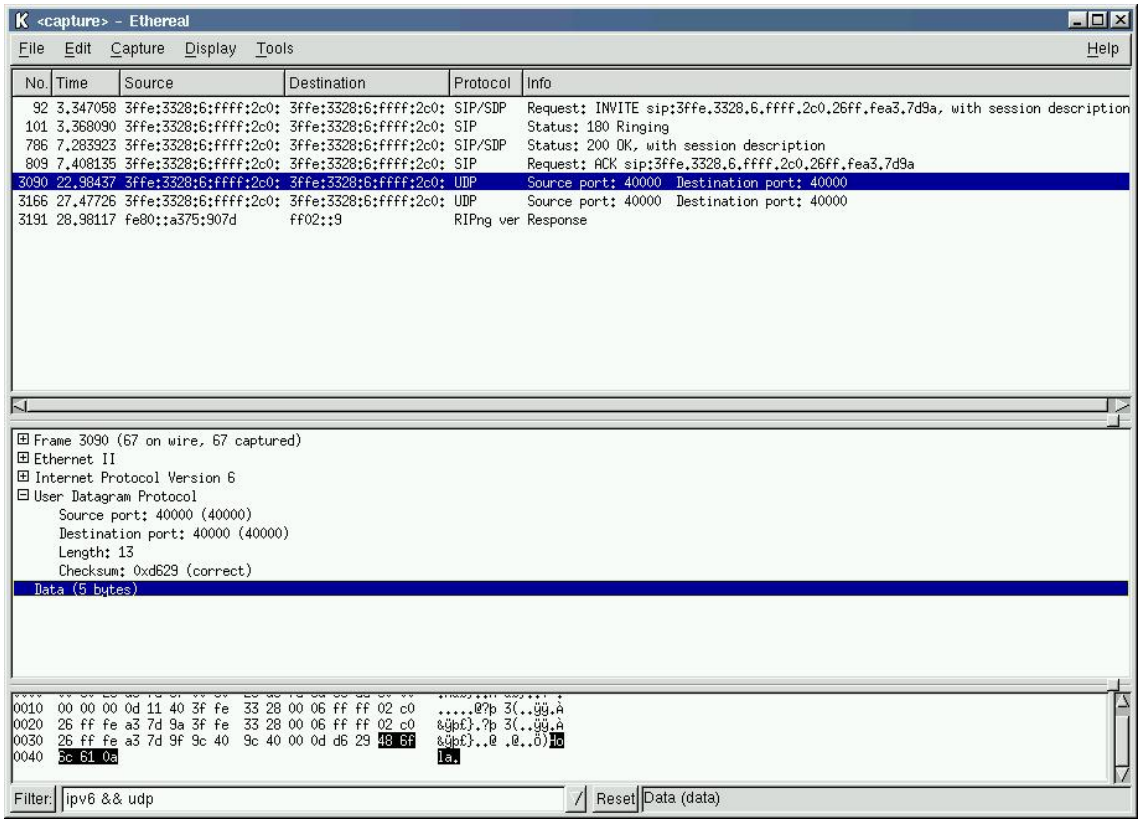


Figura IV.2 – Sesión SIP

4.4.- Proyecto MOBYDICK

Las actividades realizadas en el presente capítulo se engloban dentro de un proyecto de ámbito internacional llamado MOBYDICK. A continuación comentaremos los objetivos con que nació este proyecto.

MobyDick es un proyecto de la unión europea perteneciente al programa IST (Information Society Technologies). El proyecto comenzó el día 1 de Enero de 2001 y tiene una duración de 36 meses.

El objetivo principal de este proyecto es continuar con la evolución de la tercera generación de telefonía móvil y la infraestructura wireless (sin-hilos). El proyecto MobyDick está encargado de definir, implementar, y evaluar una arquitectura basada en IPv6 con soporte de movilidad y que aporte mecanismos de calidad de servicio. Se utilizan un conjunto de aplicaciones multimedia interactivas y distribuidas para determinar los requisitos del sistema para la verificación, validación, y demostración de la arquitectura MobyDick.

Capítulo 5 – Conclusiones y trabajos futuros

En el presente capítulos intentaremos resumir cuales han sido los objetivos alcanzados con la realización de este trabajo. Y de igual forma mostraremos los nuevos caminos de trabajo que se han abierto y que resultarían de gran interés para continuar la labor realizada en este proyecto.

5.1.- Conclusiones

Las distintas tecnologías utilizadas durante la realización de este proyecto han evolucionado según se ha desarrollado el proyecto, y si bien el protocolo de comunicaciones SIP parecía no tener mucho interés con respecto a H.323, en la actualidad se han desarrollado gran cantidad de productos y aplicaciones en el entorno SIP.

La integración de estas tecnologías en un entorno corporativo como el utilizado en el presente proyecto demuestra la viabilidad de las comunicaciones sobre protocolo IP, resultando necesario un estudio personalizado para la integración en cada caso particular.

Gracias a la realización de este trabajo se ha logrado crear una plataforma de Voz sobre IP en el Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid. En este momento es posible realizar una llamada entre dos lugares diferentes del departamento que dispongan de un cliente H.323,y de igual manera se pueden realizar llamadas al exterior gracias a la instalación de la pasarela NUCLEOX+ de TELDAT. Así mismo podemos usar los diferentes servicios prestados por el gatekeeper.

Por otro lado se dio un pequeño gran paso hacia la migración completa de la plataforma al protocolo Ipv6 portando uno de los clientes del protocolo SIP.

Esta plataforma queda disponible en el Departamento de Ingeniería Telemática para realizar labores de investigación que podrían continuar la labor realizada en el presente proyecto tal y como se indica posteriormente.

De esta forma mediante la realización de este proyecto se ha creado una base de conocimiento sobre el entorno H.323 y SIP que podríamos considerar de gran amplitud. Con la intención de que este conocimiento no se pierda con el paso del tiempo se han creado una serie de manuales que contienen la información necesario para la instalación y configuración de todos los componentes de la plataforma:

- Apéndice B - “Configuración de los clientes H.323”
- Apéndice C - “Instalación y configuración de la pasarela TELDAT”
- Apéndice D - “Instalación y configuración del gatekeeper”
- Capítulo 4 – “Guía básica de migración IPv4-IPv6”

5.2.- Trabajos futuros

Con la realización de este trabajo quedan abiertas muchas líneas de trabajo que permitirían continuar la labor iniciada con este proyecto fin de carrera. Veamos a continuación una lista de ellas:

- Resultaría muy interesante realizar medidas sobre la influencia de determinados parámetros de la plataforma en la calidad de las llamadas. Si bien en este proyecto se han realizado algunas de estas medidas, se podrían completar mediante la utilización de algún software generador de tráfico, lo cual nos permitiría una mejor caracterización de las capacidades de la plataforma.
- Para finalizar el trabajo iniciado con la migración a Ipv6 del cliente sipc, se podría intentar lograr la migración de todo la plataforma. Para ello sería necesario por un lado migrar algunas aplicaciones, y por otro conseguir una pasarela que permitiese trabajar con IPv6. Toda la labor de configuración de esta nueva plataforma resultaría muy interesante.
- Ahora que disponemos de un conocimiento más extenso del entorno H.323 se podrían repetir las medidas de calidad realizadas entre las diferentes universidades integrantes del proyecto PISCIS. De esta forma apreciaríamos en mejor medida la calidad que podemos llegar a obtener con una comunicación VoIP.
- En relación a la utilización de calidad de servicio en la plataforma podríamos intentar implementar algún protocolo de reserva de ancho de banda para el diferentes tráfico presente.
- Aprovechando la infraestructura desarrollada se pueden seguir implementando nuevos servicios sobre el gatekeeper que puedan aparecer directamente sobre la plataforma.

Referencias

A continuación presentamos la diferente bibliografía consultada para la realización de este proyecto:

Voz sobre IP

- Artículo “Desarrollo de servicios avanzados de voz sobre redes de paquetes”, M. C. Bartolomé, R. Panadero, C. García, J. Moreno, D. Fernández Jitel 2001
- Artículo “VoIP: una puerta hacia la convergencia”, Marcos Valiño García Dpto. de Lenguajes y Sistemas Informáticos Universidad de Vigo
- http://www.cisco.com/univercd/cc/td/doc/product/voice/sipsols/biggulp/bgsipov.htm
- http://www.svifsi.ch/revue/pages/issues/n013/in013Moreno.pdf
- http://www.monografias.com/trabajos3/voip/voip.shtml
- http://www.ericsson.com/review/2000_01/files/es2000013.pdf

SIP

- http://www.ietf.org/internet-drafts/draft-ietf-sip-rfc2543bis-03.txt
- http://www.anatel.net/whitepapers/MGCPWhitepaper2.pdf

IPv6

- “Basic Socket Interface Extensions for IPv6” IPv6 INTERNET-DRAFT: draft-ietf-ipngwg-rfc2553bis-03.txt
- “Unix network programming . Volume 1 . Networking APIs-Sockets and XTI”, Stevens, W. Richard. Referencia: L/S 004.451.9 UNIX STE

A continuación se agrupan diferentes direcciones web utilizadas para la descarga de software necesario en la plataforma, así como para la consulta de manuales de la pasarela:

Proyecto PISCIS	matrix.it.uc3m.es/~piscis
Proyecto Openh323	www.openh323.org
OpenGatekeeper	www.opengatekeeper.org
Microsoft Netmeeting	www.microsoft.com/netmeeting
TELDAT S.A.	www.teldat.es

Apéndice A - Presupuesto

A través de este proyecto se ha generado una plataforma que proporciona comunicación basada en Voz sobre IP, con acceso a la red de telefonía básica conmutada. Esta plataforma se podría utilizar para una explotación comercial, si bien queda más orientada a un entorno educativo en el Departamento de Ingeniería Telemática para el desarrollo y estudio de aplicaciones y servicios sobre ella.

Esta plataforma se encuentra situada en la red IP del Departamento de Ingeniería Telemática de la universidad Carlos III, utilizando diferentes equipos situados en el departamento. El diferente software empleado funciona sobre equipos de trabajo con sistemas operativos Windows 98, 2000 y NT, así como Linux SUSE y RedHat.

Posteriormente se decidió aumentar las capacidades de esta plataforma migrando algunos de los clientes disponibles desde el protocolo IPv4 hasta IPv6 cumpliendo de esta manera uno de los requisitos del proyecto MOBYDICK, dentro del cual se engloba el desarrollo de este proyecto de fin de carrera.

A lo largo del desarrollo del proyecto fueron necesarias diferentes fases de estudio sobre las diferentes tecnologías empleadas en el proyecto, así como las posteriores fases de documentación que permiten ofrecer información estructurada para todo aquel que desee informarse adecuadamente sobre los diferentes temas estudiados.

En el presente presupuesto detallamos las diferentes tareas que han compuesto el proyecto, así como los costes asociado a cada una de estas etapas, de manera que podamos valorar el esfuerzo incurrido en el desarrollo del proyecto.

A.1. Descomposición en Tareas

En el desarrollo de este proyecto se han definido las tareas que exponemos a continuación. En cada tarea especificamos los objetivos, entregas, dependencias, duración y esfuerzo. Este último esta basado en una jornada de trabajo a tiempo parcial de 4 horas/día y 20 días/mes, que son las condiciones de trabajo en que se ha completado el proyecto.

Actividad A: Montaje de una plataforma de Voz sobre IP
Actividad B: Medida de la calidad de comunicación
Actividad C: Migración a IPv6
Actividad D: Documentación y memoria técnica del proyecto

Tarea A.1: Análisis del estado del arte de VoIP
Objetivos:

- Estudio desde diferentes fuentes de información sobre las bases de las tecnología a emplear.

Entregas: Lograr una capacidad adecuada para afrontar la búsqueda de componentes en una plataforma VoIP.
Dependencias: Esta tarea comenzará una vez firmado el comienzo del proyecto.
Duración: 1 mes
Esfuerzo: 1 personas-mes

Tarea A.2: Búsqueda y análisis de aplicaciones disponibles en el mercado
Objetivos:

- Localización de los diferentes componentes que forman parte del proyecto.
- Estudio sobre el funcionamiento de estos componentes.

Entregas: Identificar los componentes que formarán parte de la plataforma.
Dependencias: Tarea A.1
Duración: 2 meses
Esfuerzo: 0.5 personas-mes

Tarea A.3: Análisis, diseño e implementación de la infraestructura

Objetivos:

- Incorporación de todos los elementos sobre la plataforma.
- Comprobación del correcto funcionamiento

Entregas: Disposición de una plataforma base para la realización de pruebas de calidad.

Dependencias: Tarea A.2

Duración: 6 semanas

Esfuerzo: 0.75 personas-mes

Tarea A.4: Documentación y memoria

Objetivos:

- Generación de la información estructurada adecuadamente para facilitar una posterior búsqueda de datos relevantes sobre la plataforma.

Entregas: Memoria donde se presente toda la información relevante.

Dependencias: Tarea A.3

Duración: 6 semanas

Esfuerzo: 0.75 personas-mes

Tarea B.1: Identificación y análisis de los parámetros de calidad

Objetivos:

- Determinar los parámetros de la plataforma VoIP que intervienen en la fijación de la calidad de la comunicación entre extremos.

Entregas: Documentación donde se reflejen los parámetros que serán estudiados en la siguiente fase.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea A.3.

Duración: 1 mes

Esfuerzo: 1 persona-mes

Tarea B.2: Desarrollo del software necesario para la medida de parámetros

Objetivos:

- Desarrollar un software adecuado que permita la medida de los parámetros descritos en el apartado anterior y que determinarán la calidad y viabilidad de una comunicación en la plataforma VoIP.

Entregas: Aplicación que permita la medida de parámetros de calidad y permita su almacenamiento para un posterior procesamiento.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea B.1.

Duración: 2 semanas

Esfuerzo: 2 persona-mes

Tarea B.3: Procesamiento de los datos obtenidos el la tarea B.2.

Objetivos:

- Procesar la información obtenida en la tarea anterior ofreciendo una información más depurada que permita un estudio de la calidad de la comunicación.

Entregas: Información depurada con un análisis y estudio sobre la viabilidad de la comunicación VoIP en la plataforma.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea B.2.

Duración: 2 semanas

Esfuerzo: 2 personas-mes

Tarea B.4: Documentación y memoria

Objetivos:

- Generación de la información estructurada adecuadamente con los resultados obtenidos en las pruebas.

Entregas: Memoria donde se presente toda la información relevante.

Dependencias: Tarea B.3

Duración: 3 semanas

Esfuerzo: 1.25 personas-mes

Tarea C.1: Análisis del estado actual sobre la transición IPv4-IPv6

Objetivos:

- Estudio desde diferentes fuentes de información sobre las bases de las tecnología a emplear.

Entregas: Lograr una capacidad adecuada para afrontar la identificación de llamadas a IPv4 y su cambio a IPv6.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea A.2.

Duración: 1 mes

Esfuerzo: 1 persona-mes

Tarea C.2: Estudio previo de la aplicación

Objetivos:

- Estudiar el código del software e identificar las componentes que hacen referencia a IPv4.

Entregas: Identificar las estructuras que será necesario modificar.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea C.1.

Duración: 1 mes

Esfuerzo: 1 personas-mes

Tarea C.3: Modificación y depuración del software

Objetivos:

- Realizar los cambios necesarios sobre el código y comprobar su validez.

Entregas: Código de la aplicación en correcto funcionamiento sobre una red IPv6.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea C.2.

Duración: 1 mes

Esfuerzo: 1 persona-mes

Tarea C.4: Pruebas sobre un escenario

Objetivos:

- Realizar los cambios necesarios sobre el código y comprobar su validez.

Entregas: Código de la aplicación en correcto funcionamiento sobre una red IPv6.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea C.3.

Duración: 2 semanas

Esfuerzo: 2 personas-mes

Tarea C.5: Documentación y memoria

Objetivos:

- Generación de la información estructurada adecuadamente para facilitar una posterior búsqueda de datos relevantes sobre la plataforma.

Entregas: Memoria donde se presente toda la información relevante.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea C.4.

Duración: 3 semanas

Esfuerzo: 1.5 personas-mes

Tarea D.1: Documentación y memoria técnica del proyecto

Objetivos:

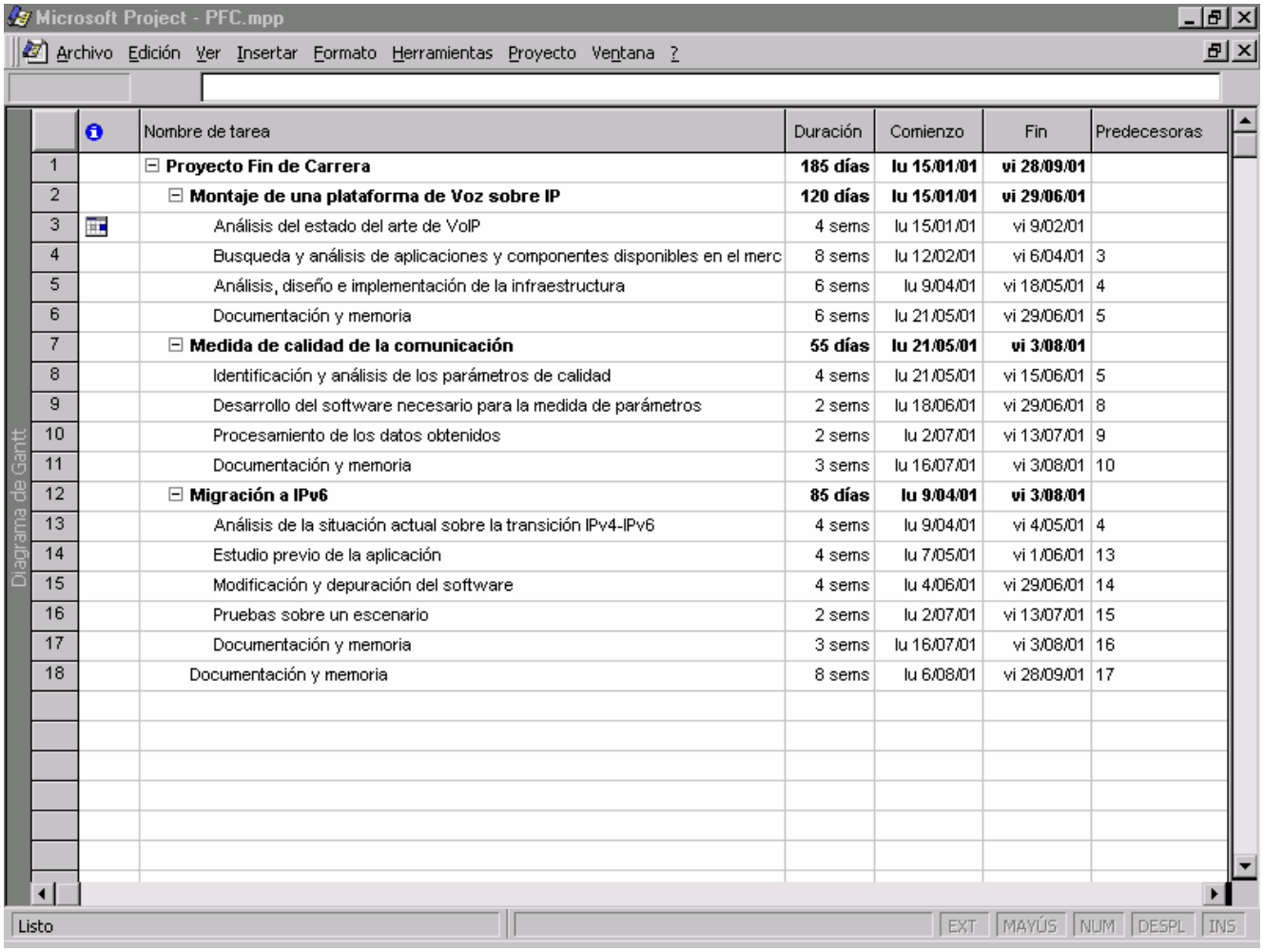
- Redactar toda la memoria referente al proyecto de fin de carrera.

Entregas: Memoria donde se presente toda la descripción del proyecto fin de carrera.

Dependencias: Esta tarea puede comenzar una vez finalizada la tarea C.5.

Duración: 2 meses

Esfuerzo: 0.5 personas-mes



	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**Proyecto Fin de Carrera**	**185 días**	**lu 15/01/01**	**vi 28/09/01**	
2	**Montaje de una plataforma de Voz sobre IP**	**120 días**	**lu 15/01/01**	**vi 29/06/01**	
3	Análisis del estado del arte de VoIP	4 sems	lu 15/01/01	vi 9/02/01	
4	Busqueda y análisis de aplicaciones y componentes disponibles en el merc	8 sems	lu 12/02/01	vi 6/04/01	3
5	Análisis, diseño e implementación de la infraestructura	6 sems	lu 9/04/01	vi 18/05/01	4
6	Documentación y memoria	6 sems	lu 21/05/01	vi 29/06/01	5
7	**Medida de calidad de la comunicación**	**55 días**	**lu 21/05/01**	**vi 3/08/01**	
8	Identificación y análisis de los parámetros de calidad	4 sems	lu 21/05/01	vi 15/06/01	5
9	Desarrollo del software necesario para la medida de parámetros	2 sems	lu 18/06/01	vi 29/06/01	8
10	Procesamiento de los datos obtenidos	2 sems	lu 2/07/01	vi 13/07/01	9
11	Documentación y memoria	3 sems	lu 16/07/01	vi 3/08/01	10
12	**Migración a IPv6**	**85 días**	**lu 9/04/01**	**vi 3/08/01**	
13	Análisis de la situación actual sobre la transición IPv4-IPv6	4 sems	lu 9/04/01	vi 4/05/01	4
14	Estudio previo de la aplicación	4 sems	lu 7/05/01	vi 1/06/01	13
15	Modificación y depuración del software	4 sems	lu 4/06/01	vi 29/06/01	14
16	Pruebas sobre un escenario	2 sems	lu 2/07/01	vi 13/07/01	15
17	Documentación y memoria	3 sems	lu 16/07/01	vi 3/08/01	16
18	Documentación y memoria	8 sems	lu 6/08/01	vi 28/09/01	17

Figura A-1 Tareas del proyecto

A.2.- Diagrama de Gantt

Para presentar la precedencia entre las diferentes tareas de manera gráfica utilizaremos un diagrama de Gantt. Mediante este gráfico podemos apreciar las dependencias entre tareas así como su disposición temporal.

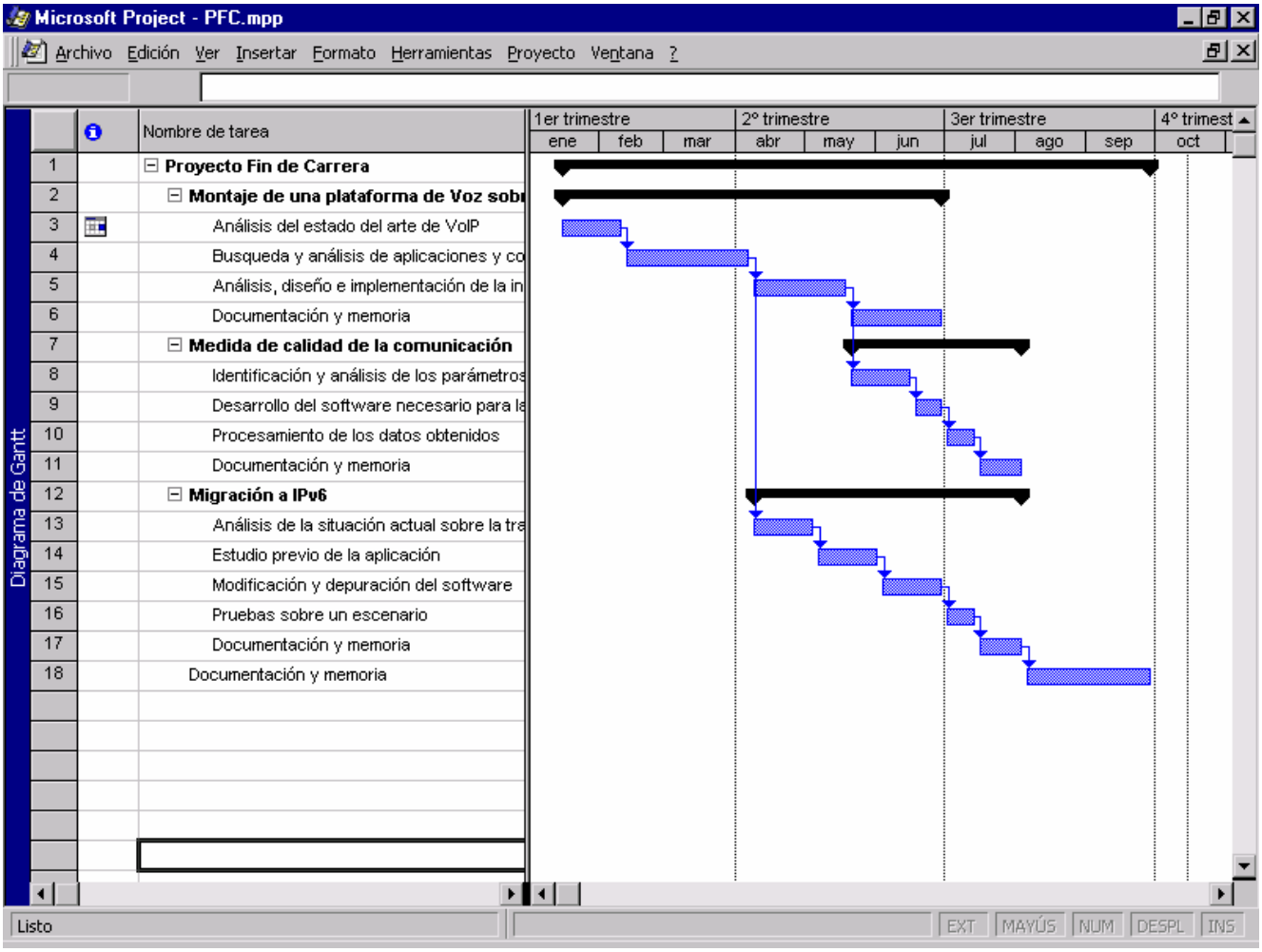


Figura A-2 Diagrama de Gantt del proyecto .

A.3.- Costes

A.3.1. Costes de personal

El proyecto tiene los costes de personal que se describen en la siguiente tabla.

Concepto	Cantidad	Precio unitario	Importe
Ingeniero de Telecomunicación	740 horas	5.500 pts/hora	4.070.000 ptas.
		Total	4.070.000 ptas.

Tabla A.1 Costes de personal.

A.3.2. Costes de Material

Concepto	Cantidad	Precio unitario	Importe
Ordenador PC multimedia	3	250.000 ptas.	750.000 ptas.
Licencias de software de desarrollo	3	100.000 ptas.	300.000 ptas.
Router Nucleox+ TELDAT	2	500.000 ptas.	1.000.000 ptas.
Material fungible	1	50.000 ptas.	50.000 ptas.
Alquiler de la línea telefónica	4 meses	1.500 ptas.	6.000 ptas
		Total	2.106.000 ptas.

Tabla A.2 Costes de material.

A.3.3. Costes Totales

Concepto	Importe
Costes de personal	4.070.000 ptas. / 24461,19 Euros
Costes de material	2.106.000 ptas. / 12657,31 Euros
Total	6.176.000 ptas. / 37118,5 Euros

Tabla A.3 Costes totales.

El presupuesto total de este proyecto asciende a **seis millones ciento setenta y seis mil** pesetas (treinta y siete mil ciento dieciocho coma cinco Euros).

Madrid, 16 de Octubre de 2001

El ingeniero proyectista

Fdo. Carlos García García

Apéndice B – Manual de instalación y configuración de los clientes H.323

A través de este manual pretendemos hacer accesible el servicio de Voz sobre IP a todo el Departamento de Ingeniería Telemática de la forma más sencilla posible. A continuación describiremos como se pueden obtener y configurar los clientes H.323 más interesantes para lograr realizar cualquier tipo de llamada.

Vamos a distinguir dos servicios diferentes, según deseemos comunicarnos con el exterior de la universidad o simplemente deseemos contactar con algún cliente H.323 dentro de la universidad.

Para conseguir la mayor penetración posible de este servicio en el departamento, comentaremos dos clientes diferentes, según el sistema operativo que estemos utilizando: Windows o Linux.

B.1.- Clientes H.323

- ***Microsoft Netmeeting***: se trata del cliente H.323 de la compañía Microsoft especialmente diseñado para sus sistemas operativos Windows 98, NT o 2000. Podemos descargar el programa desde las siguientes direcciones:

Web Microsoft	www.microsoft.com/netmeeting
Proyecto Piscis	www.matrix.it.uc3m.es/~piscis/software

Hasta la fecha la versión más reciente de este cliente es Netmeeting v3.01. Que es con la que hemos realizado todas las pruebas. La principal ventaja de este cliente es la gran cantidad de codecs que adjunta. De esta manera nos aseguramos la compatibilidad con la gateway, así como con el resto de clientes que podamos utilizar.

A continuación podemos ver la interfaz que nos ofrece este cliente H.323.

Una vez que hayamos descargado el software, el programa se instalará automáticamente y nos permitirá configurar automáticamente todos los dispositivos de audio.

Para terminar debemos seleccionar el gatekeeper que deseamos utilizar. Para ello entraremos en Herramientas → Opciones → Llamada avanzada. Aquí seleccionaremos el uso de gatekeeper pinchando sobre "Usar un equipo selector ...". La dirección que utilizaremos será 10.0.0.52[1]. A continuación debemos elegir un alias con el cual nos registraremos en el gatekeeper y será el que nos permita identificarnos para recibir llamadas. Simplemente queda pulsar sobre Aplicar y terminaremos la configuración.

En el siguiente apartado se puede revisar el ***sistema de marcado*** para comunicarse con teléfonos internos y externos.

- ***Openphone***: es el cliente H.323 para el sistema operativo Linux. Se trata de un cliente en constante desarrollo por lo que con el paso del tiempo probablemente aparezcan mejoras que en este manual no se contemplan. La versión de este software para Linux se llama Ohphone, y podemos descargarlo de las siguientes direcciones Web:

Web OpenH323	www.openH323.org/code.html
Proyecto Piscis	www.matrix.it.uc3m.es/~piscis/software

[1] En el presente capítulo, por motivos de seguridad, todas las direcciones IP han sido eliminadas o substituidas por direcciones privadas.

Este software ha sido probado bajo las distribuciones de Linux SUSE7.0 y RedHat 6.2, si bien debería funcionar correctamente para cualquier distribución.

Para una correcta instalación de este software recomendamos seguir las instrucciones de la web www.openH323.org.

Este cliente no ofrece una interfaz gráfica para el usuario del estilo de Netmeeting, y nos tendremos que conformar con introducir la información requerida a través de la línea de comandos.

Comentamos a continuación las opciones más interesantes para lograr acceder a los servicios proporcionados:
-l : coloca al cliente en modo escucha para poder recibir y realizar llamadas.
-p dirección: indica la pasarela (gateway) que utilizará.
-u nombre : será nuestro identificador cuando utilicemos el gatekeeper.
-j [# ms] : indica el tamaño del buffer de jitter en ms. Si estamos trabajando en una LAN se puede reducir al mínimo (30 ms) disminuyendo el retardo.

Una llamada típica a este programa sería:

./ohphone –l –u alias –j 30

Este programa cuando arranca se encarga de localizar un gatekeeper en nuestra zona. De esta manera no es necesario que indiquemos la dirección en que se ubica este, y nada más arrancar nos indicará que encontró un gatekeeper en el equipo escarabajo.it.uc3m.es.

Sin embargo, este cliente no resulta del todo práctica para nuestros intereses por un gran problema. Debido a tratarse de un programa de libre distribución, tan solo incorpora tres codecs muy básicos: G.711 ley_u, G.711 ley_A y GSMv6. Por desgracia ninguno de estos tres

codecs se encuentran implementados en la gateway de manera que no podremos realizar llamadas a RTC usando este cliente. Como ya hemos comentado este software se encuentra en constante desarrollo, y en el momento en que incorporen nuevos codecs su uso podría resultar muy interesante. Por el momento el soporte para nuevos codecs se puede conseguir a través de hardware específico. En concreto la tarjeta de sonido Quicknet ofrece soporte para los codecs más utilizados en telefonía IP.

En el siguiente apartado se puede revisar el ***sistema de marcado*** para comunicarse con teléfonos internos y externos.

B.2.- Sistema de marcado

Para realizar llamadas internas, es decir, a clientes H.323 que se encuentre dentro de la plataforma descrita, tenemos dos opciones:

- Si el entorno no dispusiese de un gatekeeper, podemos realizar llamadas utilizando directamente la dirección IP de la máquina destino donde se encuentra el cliente H.323.
- Si por el contrario, en el entorno existe un gatekeeper, para llamar simplemente deberemos introducir el alias con el que se encuentra escrito el cliente H.323 destino de nuestra llamada. Este alias puede ser tanto un número como un nombre (ej: 201, cgarcia, 1234, alfonso, etc.)

Para realizar llamadas a teléfonos que se encuentran fuera de la plataforma H.323 utilizaremos la gateway que hemos configurado previamente. Para ello cuando queramos llamar utilizaremos el prefijo **8**. Debemos tener en cuenta que estamos accediendo a la centralita de la universidad:

- Para llamar a cualquier teléfono perteneciente a la centralita utilizaremos la fórmula 8 + “despacho”. Ej: con **8**8756 llamaremos al despacho 4.1C01.
- En consecuencia, para llamar a un teléfono externo a la centralita, utilizaremos el prefijo (8+) **0**. Ej: **80**916119317. Atención: esta función se encuentra actualmente deshabilitada para evitar que los usuarios realicen llamadas al exterior de la universidad.

Llamadas desde fuera de la plataforma VoIP

Si queremos acceder desde el exterior de la plataforma a un cliente de esta tenemos la opción de entrar a través de la gateway. Para ello utilizaremos el acceso a RTC que consigue a través de la línea 91624**8761**. Es decir, desde cualquier teléfono de la RTC podemos marcar dicho número con el cual accederíamos a la gateway. En ese

momento aparece un tono de invitación a marcar, momento en el cual podemos introducir el número con el que el usuario destino se haya registrado en el gatekeeper.

Como hemos visto para poder recibir llamadas desde el exterior, debemos registrarnos, además de con el nombre correspondiente, con un número que será el que utilice el usuario situado en RTC para llamar a un usuario de nuestra plataforma. Para ello Netmeeting nos permite registrarnos con un alias y un número.

Apéndice C – Manual de instalación y configuración de la Gateway

En este apartado comentaremos los pasos más importantes para instalar y configurar adecuadamente la pasarela del la plataforma.

Previamente al trabajo con el equipo NUCLEOX+ de TELDAT se estuvo trabajando con una versión anterior llamada CEBRA. Ambos equipos comparten el mismo interfaz de usuario de manera que incluimos en el apartado C.1. el manual de interfaz del router CEBRA, que nos permitirá acostumbrarnos al trabajo con este tipo de equipos.

Posteriormente en los apartados siguiente se indican las variables que tendremos que conocer y modificar adecuadamente para conseguir el correcto funcionamiento de la plataforma VoIP.

C.1.- Manual de interfaz del router CEBRA

El CEBRA es un equipo desarrollado por TELDAT pensando fundamentalmente en su adaptación a los escenarios que se presentan en las redes de teleproceso más habituales. A través de sus conexiones LAN-WAN se ofrece la funcionalidad de ROUTER IP, soportando OSPF o routing dinámico.

Este router es capaz de gestionar las siguientes redes:

- LAN: Ethernet, Token Ring, LLC, ARP.
- RDSI: X.25 sobre canal D.
- WAN: Frame Relay, X.25, PPP, SDLC, etc.

C.1.1.- Descripción del interfaz de usuario

Cuando nos conectamos al router TELDAT obtenemos la siguiente pantalla:

Teldat S.A. (c)1996
Router modelo NUCLEOX-PLUS SPU M68360 N/S:XXXX/XXXX
1 LAN, 2 Lineas WAN, 2 Lineas RDSI

*

El router CEBRA nos ofrece una estructura en árbol para acceder a la configuración del equipo. Comenzamos en el proceso 1 (Gestión de consola), a partir del cual poemos acceder a los siguientes subprocesos:

Proceso 1: Gestión de consola – Su misión es facilitar el acceso a los edmás procesos otorgánoles la consola.
Proceso 2: Visualización de Eventos – Recibe mensajes del Sistema de Registro de Ebentos y los prensenta en el terminal, de acuerdo con los criterios de selección del usuario.
Proceso 3: Monitorización del estado y estadísticos – Facilita la configuración de varios parámetros, tales como dirección de red y eventos. Proporciona el acceso a la configuracioón de porotocolos, que permiten configurar sus parámetros de protocolos.
Proceso 4: Configuración – Permite al usuario monitorizar el estado y los estadísticos del hardware y software del router. Facilita el acceso a los menús de los protocolos e interfaces, que permiten al usuario monitorizar protocolos configurados y otros parámetros.

Para entrar en cualquiera de estos procesos simplemente introducimos: ***P****ROCESO pid*, donde pid es el número de proceso. Una vez que entremos en cualquiera de estos procesos, la única forma de retornar al Gestor de Consola es pulsando la tecla de escape, que por defecto es ***Ctrl+p***.

C.1.2.- Comandos básicos

? Ayuda

Presenta los comandos disponibles desde el prompt actual. También se puede teclear ? después de un **comando específico** para listar sus opciones.

Proceso

Permite el acceso a otro proceso del equipo, tal como MONITOR, VISEVEN o CONFIG. Para regresar al proceso inicial introduzca el carácter de escape (por defecto Ctrl+p).

Info-Pro

Presenta el identificador (pid) de cada proceso.

Listar

Proporciona información sobre la configuración actual durante la configuración de un interfaz.

Car-escape

Permite cambiar el carácter de escape de los procesos.

Guardar

Dentro del proceso de configuración guarda las variaciones que hallamos realizado sobre la configuración. Antes de la utilización de este comando las modificaciones no tendrán efectos.

Reiniciar

Reinicia el router CEBRA, lo cual conlleva:

Pone los contadores de software a cero

Hace un test de las redes conectadas

Borra las tablas de routing

Descarta todos los paquetes hasta que el reinicio se completa

Ejecuta el software actual

Recuerde que debe reiniciar el equipo cada vez que cambie la configuración del equipo.

Fin-conexion

Termina la conexión Telnet establecida con el equipo sin necesidad de usar ningún comando del cliente Telnet.

C.1.3.- Configuración inicial a través de un terminal serie

Para comenzar la configuración necesitaremos una conexión serie a través del terminal que aparece en el frontal del equipo, mediante un cable serie RS-232. Debemos establecer la conexión con las siguientes características: terminal asíncrono a 9600 bps sin paridad con 8 bits. Necesitaremos un programa del estilo del Hyperterminal de Windows.

Si hemos seguido estos pasos correctamente, lograremos establecer una conexión con el router, que nos presentará una pantalla como la siguiente:

Teldat S.A. (c)1996
Router modelo NUCLEOX-PLUS SPU M68360 N/S:XXXX/XXXX
1 LAN, 2 Lineas WAN, 2 Lineas RDSI
*

El símbolo '*' es el prompt (que se sustituirá por el nombre del equipo cuando lo configuremos). De esta manera entramos en el menú principal, llamado Gestor de Consola (proceso 1).

Lo primero que debemos hacer es configurar la conexión a través de la red LAN (Ethernet o Token Ring) para poder configurar el router sin necesidad de establecer una conexión por el terminal serie. Para ello debemos configurar el interfaz con la red LAN y el protocolo IP.

Desde el Gestor de Consola (prompt *) introducimos '**p 4**' para acceder al proceso de configuración. Ahora debemos configurar la interfaz 0 que se corresponde con la red LAN. Introducimos '**interfaz 0**' y en función de la LAN que tengamos (Ethernet o Token Ring) obtendremos el prompt correspondiente.

Para el caso Ethernet tendremos (ETH Config>), con '**listar todo**' obtenemos la configuración actual que compararemos con las características de nuestra LAN. Para

cambiar la dirección MAC introducimos '**Configurar Direccion-Mac**'. Finalmente volveremos al proceso de Configuración introduciendo '**salir**'.

Ahora debemos configurar el protocolo IP. Entraremos con '**p 0**' (donde 0 se corresponde con el protocolo IP. Lo primero que debemos hacer es asignar una dirección IP al interfaz 0 (red LAN). Para ello introducimos '**Conf IP> Agregar Direccion "nº interfaz" "dir IP"** '. Además configuraremos la dirección IP interna, así como la dirección IP por defecto: '**Conf IP>Configurar Dir-IP-Interna "dir IP"**' y '**Conf IP>Configurar Router-ID "dir IP"**'.

Finalmente salvaremos la configuración. Para ello volvemos al proceso de Configuración ('**salir**'), y aquí tecleamos '**guardar**'. Nos aparecerá un mensaje de confirmación, al cual respondemos si con una 's' (no con 'y'es). Para terminar reiniciamos el equipo desde el proceso Gestor de Consola (**ctrl+p**) pulsando '**reiniciar**'.

Desde este momento no necesitaremos el cable serie para configurar el equipo. Simplemente desde linux, Windows o MS-DOS llamamos a Telnet, indicando la dirección IP que hemos asignado al interfaz 0 (LAN). Con lo que conseguiremos la misma pantalla de presentación anterior.

Resumen:

** p 4*
Config> interfaz 0
ETH Config> configurar direccion-mac 40:00:00:00:00:01
ETH Config> salir
Config> protocolo ip
Config IP> agregar direccion 0 163.117.144.223 255.255.255.0
Config IP> Configurar dir-ip-interna 163.117.144.223
Config IP> Configurar router-id 163.117.144.223
Config IP> salir
Config> guardar
Config> Ctrl+p
** reiniciar*

C.1.4.- Estructura interna de interfaces

El router establece interfaces numerados para controlar todas estas puertas. Podemos ver las interfaces configuradas tecleando '**Config> listar interfaces**':

Config>
Config> Listar interfaces

Con	*Ifc*	*Tipo de interfaz*	*CSR*	*CSR2*	*Int*
---	*1*	*Router->Nodo*	*0*		*0*
---	*2*	*Nodo->Router*	*0*		*0*
LAN	*0*	*Ethernet*	*9000000*		*1C*
Linea1	*3*	*X25*	*F001600*	*F000C00*	*9E*
Linea2	*4*	*X25*	*F001620*	*F000D00*	*9D*
RDSI1	*5*	*RDSI Canal D*	*A000000*		*1B*

Config>

Podemos observar que la interfaz 0 se corresponde con la LAN, y aparece un elemento llamado Nodo con el que el router comparte dos interfaces (entrada y salida).

El nodo controla los siguientes interfaces: Nodo->Router, los X.25, y los RDSI (que transporten X.25). El resto de interfaces pertenecen al router.

Para entrar en cada interfaz, ya sea para configuración o monitorización, simplemente debemos introducir '**interfaz [número]**'. Y para regresar al menú anterior utilizaremos '**Salir**'.

C.1.4.1.- Ethernet

Para acceder a la configuración de este interfaz, debemos entrar en el proceso de configuración (*p 4) y posteriormente introducir: **Config> Interfaz 0**.

Comandos de configuración:

Tipo-conector

Asigna el tipo de conector. Los tipos posibles son: AUI (10Base5), RJ45 (10BaseT) y AUTO.

Encapsulado-ip

Selecciona el modo de transportar IP en el campo de datos de las tramas Ethernet. Los formatos posibles son: Ethernet (Ethernet tipo 8137) o IEEE-802.3 (Ethernet 802.3 puro sin 802.2).

Listar

Muestra la información actual del interfaz Ethernet.

LLC

Muestra el prompt de la configuración LLC (LLC Config>). Se requiere la configuración LLC para pasar tramas sobre la red SNA

Dentro la configuración LLC podemos indicar la dirección MAC, para ello disponemos del comando '**LLC Config>Configurar Direccion-mac**'. Podemos indicar una dirección localmente administrada como 40:00:00:00:00:01, o una dirección globalmente administrada en cuyo caso debe empezar por 00:05:64 como por ejemplo 00:05:64:00:00:01.

Es interesante distinguir las dos nomenclaturas existentes para este tipo de direcciones.

Formato Token Ring – separado por dos puntos. Ej: 00:05:64:00:00:80

Formato canónico o Ethernet – separado por el guión. Ej: 00-A0-26-00-00-01

Ambas direcciones indican la misma dirección Mac.

Salir

Vuelve al prompt de configuración Config>.

Ejemplo de configuración Ethernet:

** p 4*

Config> int 0

Config Ether> Configurar Direccion-mac 40:00:00:00:00:01

Config Ether> listar todo

Config Ether> salir

Config> guardar

Config> Ctrl+p

** reiniciar*

C.1.4.2.- TCP/IP

Para acceder a la configuración de este protocolo, debemos entrar en el proceso de configuración (*p 4) y posteriormente introducir: **Config> Protocolo IP**.

Comandos de configuración:

Agregar

Control-Acceso – Permite especificar la clase de paquetes que hay que hacer progresar o descartar según el tipo de entrada

Dirección – Añade una dirección IP a uno de los interfaces hardware del router. Un interfaz no recibirá o transmitirá paquetes hasta tener al menos una dirección IP.

Sintaxis: Conf IP>Agregar Direccion <nºinterfaz, direccion-IP, mascara-IP>

Ej: Conf IP> Agregar Direccion 0 163.117.144.210 255.255.255.0

Filtro – Designa un filtro para una red o subred IP. Un paquete IP que cumple unas condiciones de filtrado no será encaminado, y simplemente será rechazado.

Ruta – Añade rutas estáticas IP de red o subred a la tabla de encaminamiento.

Cambiar

Dirección, Filtro, Ruta – Permite cambiar los valores que habíamos introducido anteriormente.

Borrar

Control-Acceso, Dirección, Defecto, Filtro, Ruta – Permite borrar los parámetros establecidos.

Habilitar/Deshabilitar

Broacast-directo – controla el progreso de aquellos paquetes IP cuyo destino es una dirección broadcast de red no local.

Multicamino-por-paquete – En caso de dos posibles rutas de igual coste se encamina de acuerdo a una cola circular (Round-Robin).

Listar

Todo, Controles-acceso, direcciones, Protocolos, Rutas, Longitudes - Muestra la información actual de configuración IP.

Salir

Vuelve al prompt de configuración Config>.

Mover

Control-acceso – Permite cambiar el orden de la lista ded control de acceso.

Configurar

Control-acceso – Activa y desactiva el control de acceso IP

Dir-broadcast – Especifica el formato de las direcciones Broadcast

Long-cache – Configura el número máximo de entradas en el cache de routing IP.

Defecto – Establece una ruta por defecto

Dir-IP-interna – Establece la dirección IP interna como equipo, no la asociada a un interfaz.

Router-ID – Establece la dirección IP por defecto que el router utilizará cuando genere varios tipos de tráfico IP.

Ejemplo de configuración TCP/IP:

** p 4*
Config> protocolo ip
Config IP> agregar direccion 0 163.117.144.223 255.255.255.0
Config IP> Configurar Router-ID 163.117.144.223
Config IP> Configurar Dirreccion-ip-interna 163.117.144.223
Config IP> listar todo
Config IP> salir
Config> guardar
Config> Ctrl+p
** reiniciar*

C.1.4.3.- Protocolo PPP

PPP son las siglas de Point-To-Point Protocol. Proporciona un mecanismo para transmitir datagramas de diversos protocolos sobre un enlace punto a punto.

Asignación de interfaz PPP

Interfaz serie como interfaz PPP Síncrono:

Config> asignar enlace ppp
Que línea desea cambiar? [0]? 1
Config> interfaz 1

Interfaz serie como interfaz PPP Asíncrono:

Config> asignar enlace asppp
Que línea desea cambiar? [0]? 2
Config> interfaz 2

Interfaz PPP sobre un acceso básico RDSI:

Config> agregar interfaz ppp

Escriba acceso básico RDSI (1,2): 1

Agregado interfaz PPP-DIAL con num: 5

Config> interfaz 5

Interfaz PPP sobre interfaz ed comandos AT:

Config> agregar interfaz atppp

Escriba puerto DTE (1,2): 2

Agregado interfaz ATPPP-DIAL con num: 7

Config> interfaz 7

El protocolo que es soportado sobre el interfaz PPP es el IP. Será necesario activar IP sobre el interfaz PPP y asignar una dirección IP al citado interfaz.

**p 4*

Configuración de usuario

Config>p ip

Configuración del protocolo IP

Conf IP>agre dir

*Para que ifc es la dirección? [0]? (*podemos consultarlo con *Config>listar int)*

Nueva dirección [0.0.0.0]? 194.6.5.1

Mascara [255.255.255.0]? 255.255.255.0

Conf IP> salir

Config> guardar

Desea guardar la configuración? S

Guardando la configuración ...OK

Config> Ctrl+p

** Reiniciar*

Comandos de configuración del interfaz PPP

Listar

Todo, línea, LCP, NCP, IPCP, Autenticación, Facilidades.

Configurar

Línea – reloj, codificación, inactivo, longitud de trama, **velocidad de línea**, retardo de transmisión.

LCP – opciones, parámetros.

NCP

IPCP

Autenticación

Habilitar/Deshabilitar – NAT, autenticación.

Salir

Ejemplo de configuración PPP sobre RDSI:

** p 4*

Config> agregar interfaz ppp

Escriba acceso básico RDSI (1,2): 1

Agregado interfaz PPP-DIAL con num: 2

Config> interfaz 2

Config circuito> configurar direccion-destino 3016

Config circuito> configurar tiempo 60

Config circuito> habilitar salientes

Config circuito> habilitar entrantes

Config circuito> salir

Config> guardar

Config> Ctrl+p

** reiniciar*

Ejemplo de configuración PPP sobre Modem (Serie):

** p 4*

Config> agregar interfaz atppp

Escriba acceso básico DTE (1,2): 2

Agregado interfaz PPP-DIAL con num: 4

Config> interfaz 4

Config circuito> configurar direccion-destino 2016

Config circuito> configurar tiempo 60

Config circuito> habilitar salientes

Config circuito> habilitar entrantes

Config circuito> encap

PPP Config> configurar linea velocidad 57600

PPP Config> salir

Config circuito> salir

Config> guardar

Config> Ctrl+p

** reiniciar*

C.2.- La pasarela NUCLEOX+

Para una correcta configuración de las Gateways debemos destacar tres tablas donde se almacena diferente información.

- **Tabla de direcciones (Table Address):** se trata de una tabla de routing que la gateway consulta cuando desea alcanzar una dirección destino. A continuación podemos apreciar la actual configuración para Piscis2.

```
PISCIS_2 H.323 Config>list ta ad

                          STRIP      DIAL-OUT  CODEC      TECH
Ord       IP.Add     TELEPHONE PREFIX       PREFIX  CLASS     PREFIX

1       138.4.2.48        101    0              --
2       138.4.2.48        102    0              --
3       138.4.2.48        104    0              --
4      138.4.50.24        111    0               --
5      138.4.50.24        112    0               --
6      138.4.50.23        115    0               --
7      138.4.50.20        116    0               --
8   163.117.139.168       211    0                --
9   163.117.139.161       201    0                --
10  163.117.139.157       205    0                 --
11  163.117.139.140       206    0                 --
12  163.117.139.168         7    0                --
13      138.4.2.48          8    0              --
```

Como podemos apreciar aparecen configurados todos los números de la red PISCIS que hemos definido. Así como los prefijos 7, 8 y 9 para salir a la RTC a través de las diferentes universidades.

Para añadir nuevas direcciones se utiliza la instrucción:

PISCIS_2 H.323 Config> ADD ADDRESS

- **Tabla de prefijos (Table Prefix):** contiene diferentes prefijos con la longitud total de cada teléfono, que nos permite no utilizar la almohadilla (#) al terminar de marcar el número. A continuación podemos ver un ejemplo de configuración de Piscis2.

```
PISCIS_2 H.323 Config>list ta pr

Order  PREFIX        LENGTH

 1     20            3
 2     21            3
```

Mediante esta tabla cuando marquemos 211, 212 o 201, no tendremos que pulsar #. Para añadir nuevos prefijos se utiliza la instrucción:

PISCIS_2 H.323 Config> ADD PREFIX

- **Tabla de líneas (Table line):** aquí aparecen los números de teléfono asociados a cada línea. De manera que si llega alguna llamada con destino uno de estos números, se encamina directamente. Veamos la configuración de Piscis2.

```
PISCIS_2 H.323 Config>list ta li

Order LINE      TELEPHONE  STRIP-PREFIX DIAL-OUT-PREFIX
```

```
1    1          211      0
2    2          212      0
3    3          213      0
4    4          214      0
5    4            7      1
6    4    916245949      0
```

Aquí aparecen las diferentes extensiones asociadas a cada línea. También se debe incorporar el número prefijo utilizado para salir a RTC.

El prefijo utilizado para salir a RTC debe aparecer en Table Line y en Table Address

Para añadir una nueva línea se utiliza la instrucción:

PISCIS_2 H.323 Config> ADD LINE

Existen otra serie de parámetros que se deben configurar para un correcto funcionamiento. Los enumeramos todas a continuación:

- Configuración de la dirección interna de la Gateway, dirección del Gatekeeper y zona del gatekeeper. Veamos la configuración de Piscis2.

```
PISCIS_2 H.323 Config>list gw
Gateway internal address: 163.117.139.168
Fast Connect: Enabled          Q931 port: 1720
```

```
H.323 mode: Compatible          UDP  port: 20000

Gatekeeper address 163.117.139.157   Gateway name: piscis2
Gatekeeper zone: Opengate        Tech-Prefix :
                        Register E.164: Enabled

RAS port: 1719              RAS time to live: 60
RAS timeout: 20             RAS Connection attempt fail:  10

Enable Service Addr 0.0.0.0      Type of Service Disable: Disable Lines
```

La zona del Gatekeeper debe coincidir exactamente con la especificada en nuestro GKR.

Para especificar esta serie de parámetros se utilizan las instrucciones:

PISCIS_2 H.323 Config> SET GATEKEEPER ADDRESS

PISCIS_2 H.323 Config> SET GATEKEEPER ZONE

PISCIS_2 H.323 Config> SET GATEWAY ADDRESS

PISCIS_2 H.323 Config> SET GATEWAY NAME

- Para cada línea se debe especificar el número de teléfono asociado, un identificador H.323 (con el que se registrará en el GKR), tipo de interfaz, codec utilizado, y el nº de tramas H.323 por paquete RTP. Veamos la configuración de Piscis2 para un teléfono y para la RTC.

```
PISCIS_2 H.323 Config>list li 1
Telephone number: 211          Interface type: FXS
Direct dialing:                State: Enabled
```

```
Identifier H.323:  211              Priority: 9

Codec: G723 6.4Kbps                 VAD: Disabled
Frames H.323/packet RTP: 2 ( 48 bytes)  DTMF relay : in band

Speaker attenuation: 0   dB         Tone level: 3 dB
Mic gain: 10 dB
```

```
PISCIS_2 H.323 Config>list li 4
Telephone number: 214           Interface type: FXO
Direct dialing:                 State: Enabled
Identifier H.323:  214              Priority: 9

Codec: G723 6.4Kbps                 VAD: Disabled
Frames H.323/packet RTP: 2 ( 48 bytes)  DTMF relay : in band

Speaker attenuation: 0   dB         Tone level: 3 dB
Mic gain: 10 dB
```

Podemos apreciar que la conexión a la RTC es una interfaz de tipo FXO, mientras que la conexión a un teléfono es de tipo FXS.

Para un correcto funcionamiento con Netmeeting se debe especificar Frames H.323/packet RTP=2

Para especificar esta serie de parámetros se utilizan las instrucciones:

PISCIS_2 H.323 Config> SET LINE CODEC

PISCIS_2 H.323 Config> SET LINE FRAMES/PACKET

PISCIS_2 H.323 Config> SET LINE IDENTIFIER

PISCIS_2 H.323 Config> SET LINE TELEPHONE-NUMBER

PISCIS_2 H.323 Config> SET LINE TYPE interface

Finalmente, si deseamos **monitorizar** el estado del Gateway y los paquetes entrantes y salientes debemos seguir los siguiente pasos:

- Entrar en modo "ELS MONITOR" de P 3 escribiendo:

  PISCIS_2+ event

- Desde este modo escribir:

  PISCIS_2 ELS> ENABLE TRACE SUBSYSTEM H.323 ALL

- Finalmente entrar en *P 2*

Para que todas las modificaciones realizadas entren en funcionamiento escribir "PISCIS_2 H.323 Config>APPLY".
Grabar la configuración "PISCIS_2 Config>SAVE".
Y finalmente resetear la máquina "PISCIS_2 *RESTART"

Manual del teléfono de la gateway

La plataforma de Voz sobre IP que hemos creado incorpora un teléfono en una de las interfaces de la gateway para poder comunicarse desde el centro de cálculo. El sistema de marcado será el mismo que el descrito anteriormente con algunas salvedades.

Debido a que solo podemos introducir números a través del teléfono, si queremos contactar con algún usuario dado de alta en el gatekeeper, este deberá utilizar un número como alias.

Apéndice D – Manual de instalación y configuración del gatekeeper

El gatekeeper que hemos montado en la plataforma es el desarrollado por el grupo de desarrollo “OpenH323 Proyect”. Se trata de un software libre que ofrecen a través de su página web.

OpenGateKeeper	*http://www.opengatekeeper.org/download.htm*

En esta página también se encuentran ejemplos tanto del registro de configuración para Windows como de el archivo de configuración opengate.ini para el

sistema operativo Linux, ambos necesarios para el correcto funcionamiento del gatekeeper.

El software que integra el Gatekeeper necesita la instalación de una serie de bibliotecas de funciones para poder compilar el código y obtener el archivo ejecutable. Las bibliotecas correspondientes son PWLib y OpenH323, ambas disponibles en el siguiente enlace:

PWLib **OpenH323**	*http://www.openH323.org/code.html*

A través de este enlace podemos descargar tanto el código fuente como los correspondientes archivos ejecutables. Así mismo, encontraremos este software disponible para los sistemas operativos Windows y Linux. En nuestro caso optamos por utilizar Linux para maximizar la estabilidad de la plataforma y permitir el desarrollo de nuevos servicios en el Gatekeeper gracias al concepto de software libre.

En la siguiente tabla podemos apreciar las diferentes versiones del software que finalmente se ha integrado en la plataforma:

Biblioteca PWLib	pwlib_min_1.1pl15
Biblioteca OpenH323	openh323_1.1beta1
Software Gatekeeper	opengate_0.7alpha0

D.1.- Instalación del gatekeeper

A continuación se enumeran los pasos que se deben seguir para descargar, compilar e instalar el software del gatekeeper:

1. Para comenzar tendremos que descargar las fuentes de la biblioteca OpenH323 y compilarlas de manera que posteriormente puedan ser enlazadas con el código de la aplicación. Como ayuda podemos seguir los pasos descritos en el enlace http://www.openH323.org/build.htm
2. Ahora debemos descargar el código fuente de la aplicación a través del enlace comentado anteriormente.
3. Para compilar el código utilizaremos el compilador de lenguaje C para Linux *gcc*, así como la herramienta *make*:
    a) Los archivos con el código fuente deben localizarse en un subdirectorio de OpenH323 llamado opengate.
    b) A continuación crearemos las dependencias usando los comandos “make optdepend” y “make debugdepend”.
    c) Y finalmente compilaremos usando “make opt” para la versión ejecutable o “make debug” para la versión de depuración.

4. Las herramientas de compilación son gcc version 2.95 o posterior para Linux.
5. A partir de las compilaciones realizadas hemos comprobado que no aparece ningún problema cuando se utiliza una distribución “Red Hat 6.2”, si bien cuando utilizamos “Suse” aparecen algunos problemas con la biblioteca “visón”. Podemos encontrar la ayuda necesaria para resolver estos problemas en http://www.openH323.org/bison_bug.html.

A continuación comentaremos los errores que se cometen con mayor frecuencia con la intención de agilizar este proceso.

1. Si hemos descargado el software en un directorio que no cuelga del directorio raiz deberemos indicarlo a través de una variable de entorno. En concreto deberemos modificar *OPENGATEDIR=$HOME/...*
2. De igual forma tendremos que indicar la posición de las bibliotecas PWLib y OpenH323 a través de variables de entorno.

Pasos para ejecutar el GKR

Aunque hemos decidido utilizar Linux como sistema operativo comentaremos los pasos para el sistema operativo Windos por si resultara necesario utilizarlo en algún momento.

1. Sobre Windows NT:

    a. Este software funciona como si fuera un servicio por lo que resulta necesario ser Administrador para poder ejecutar la aplicación.
    b. Para instalar el servicio usaremos el comando “opengate install”, y para arrancarlo será necesario utilizar “opengate start”.
    c. Si deseamos que el programa nos ofrezca información de depuración durante la ejecución podemos arrancar el Gatekeeper en modo depuración con el comando “opengate debug”. En este caso podemos modificar el nivel de información que deseamos obtener.

2. Sobre Linux:

    a. En este caso podremos encontrar el correspondiente archivo ejecutable en el directorio obj_linux_x86_d (si deseamos correr la aplicación como demonio) o en obj_linux_x86_r (para el ejecutable). Para obtener información con todas las opciones que permite el Gatekeeper podemos utilizar “opengate –h”.

D.2.- Configuración del GKR

En el caso de la versión para Windows la información de configuración del Gatekeeper será accesible a través del registro de Windows. En concreto la clave de registro que contiene esta información es:

HKEY_LOCAL_MACHINE\Software\Egoboo\opengate\CurrentVersion

Para la versión Linux, podremos modificar esta información mediante un archivo de configuración que debería localizarse en ~/.pwlib_config/opengate.ini.

A través de la configuración del Gatekeeper podremos controlar el comportamiento del gatekeeper. Las diferentes opciones de configuración se dividen en varias secciones:

1) System
    a. Log Level
    b. IsGKRouted
    c. Route H245
    d. Local Addrss
    e. Gatekeeper Id
    f. Endpoint TTl
    g. Max Bandwidth
    h. Min Call Bandwidth
2) RASLog

   Controla los mensajes Log
    a. File
    b. Level
3) Neighbours

Lista de gatekeepers con los que puede comunicarse para identificar terminales.

4) Prefixes

A continuación comentamos los parámetros que han resultado más interesantes para la configuración de nuestra plataforma:

a. IsGKRouted: indica si deseamos que la señalización de la llamada pase por el gatekeeper o no.
b. Gatekeeper Id: permite configurar en las gateways la zona del gatekeeper. Resulto necesario cambiarla con respecto a la dada por defecto ya que las pasarelas Teldat sólo permitían nombres de 8 caracteres y no admitía espacios en blanco. Finalmente se ha obtado por denominar a la zona con el nombre “Opengate”.
c. Log Level: permite especificar el nivel de información de depuración que se desea recibir. Seleccionando el nivel 3 lograremos la máxima información posible.

Podemos apreciar otras diferentes configuraciones en ejemplos que se proporcionan en la página web del grupo OpenGateKeeper:

Windows	http://www.opengatekeeper.org/opengate.reg
Linux	http://www.opengatekeeper.org/opengate.ini

Para obtener información actualizada sobre la configuración del gatekeeper podemos consultar el siguiente enlace *http://www.opengatekeeper.org/settings.htm*.

Configuración de prefijos en el Gatekeeper

Con la intención de facilitar el uso de la pasarela configuraremos un prefijo en el gatekeeper de manera que todo número de destino que comience por 8 será reencaminado a la centralita de la universidad a través de la gateway.

De esta manera desde nuestra plataforma de voz sobre IP podremos ponernos en contacto con cualquier punto de la universidad aunque este no disponga de un cliente H.323. La intención final es la implantación de la plataforma a nivel global, de manera que este tipo de comunicaciones serán un paso intermedio. Además dotamos a la plataforma de mayor cobertura.

La configuración de prefijos en el gatekeeper se realiza a través del archivo de configuración opengate.ini. En nuestro caso, para identificar el prefijo 8 y fijar como destinataria la gateway con dirección IP 10.0.0.234 utilizaríamos las siguientes líneas:

```
[Prefixes]
10.0.0.234=8
```

Debemos tener en cuenta que la gateway no permite números de más de 5 cifras con el prefijo 8. Esta medida se tomo para evitar que los usuarios realicen llamadas al exterior a través de la centralita de la universidad. Solo se permite la realización de llamadas a la universidad con teléfonos de 4 cifras.

Apéndice E – Aplicación Pingv4

A continuación se adjunto el código fuente de esta aplicación desarrollada para la medida del tiempo de retardo entre dos equipos cualquiera a través del protocolo UDP.

E.1.- Código fuente: Pingv4.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/time.h>


#define UDP 17
#define DEPURAR 0 // Para poner los mensajes de depuración -> 1
#define mensaje(s) {if (DEPURAR) printf(s);}

/* ------------------------------------------------------------------
 * Programa:  pingv4.c
 *
 * Propósito: medida del retardo entre dos equipos
 *
 * Autores:    Carlos García García
 *
 * ------------------------------------------------------------------
 */
void servidor(int puertoUDP);
void cliente(int puertoUDP, char host[50]);

int main(int argc, char *argv[])
{
  int puertoUDP;
  char host[50];

  if (argc<2) {
    printf(" Numero de argumentos incorrecto.\n PINGv6 [-s o -c] [n puerto] [host(dir
IP)]\n");
    exit(1);
  }

  if (argc>2) puertoUDP=atoi(argv[2]);
  else puertoUDP=6002;

  if (argc>3) strcpy(host, argv[3]);
  else strcpy(host, "163.117.139.54");

  if (strcmp(argv[1],"-s")==0) servidor(puertoUDP);
  else if (strcmp(argv[1],"-c")==0) cliente(puertoUDP, host);
  else {
    printf(" Numero de argumentos incorrecto.\n PINGv6 [-s o -c] [n puerto] [host(dir
IP)]\n");
    exit(1);
  }

  return 1;
}

void cliente(int puertoUDP, char host[50])
{
```

```
        int s_udp, estado, tiempo, quien;
        char buf[100];
        struct sockaddr_in miaddr;
        struct timeval inicio, fin, tv;
        fd_set rfds;



        memset ( &miaddr, 0, sizeof(miaddr));  /* pone a cero estructura */
        miaddr.sin_family = AF_INET;
        miaddr.sin_port = htons(puertoUDP);
        miaddr.sin_addr.s_addr = INADDR_ANY;

        mensaje(" Cliente PINGv6 lanzando conexion al servidor\n");

        if ((s_udp=socket(AF_INET, SOCK_DGRAM, UDP))==-1) {printf("Error en
socket\n");exit(1);} // CONTROL de errores
        if (bind(s_udp, (struct sockaddr *) &miaddr, sizeof(miaddr))==-1) {printf("Error en
bind\n");exit(1);}


        mensaje(" Conexión establecida. Mandando mensaje ...\n");
        sprintf(buf,"PINGv6(CARLOS)");


        memset( &miaddr, 0, sizeof(miaddr));
        miaddr.sin_family = AF_INET;
        miaddr.sin_addr.s_addr = inet_addr(host);
        miaddr.sin_port = htons(puertoUDP);


        gettimeofday(&inicio, NULL);
        estado=sendto(s_udp, buf, sizeof(buf), 0, (struct sockaddr *) &miaddr, sizeof(miaddr));
        if (estado==-1) {printf("Error en sendto\n");exit(1);}
        mensaje(" Mensaje enviado. Esperando confirmacion ...\n");


          FD_ZERO(&rfds);
          FD_SET(s_udp, &rfds);
          tv.tv_sec=5;
          tv.tv_usec=0;

          // Mediante select esperamos que llegue algo al socket UDP o pasen 5 segundos

          if ((quien=select(s_udp+1, &rfds, NULL, NULL, &tv))==-1) {printf("Error en
select\n");exit(1);}

          if (quien) {
            if (FD_ISSET(s_udp, &rfds)) {
              estado=recvfrom(s_udp, buf, sizeof(buf), 0, NULL, NULL);
              if (estado==-1) {printf("Error en recvfrom\n");exit(1);}
              mensaje("Mensaje de confirmacion recibido. Comprobando firma ...");

              gettimeofday(&fin, NULL);

              if (strncmp(buf, "CONFIRMADO", 10)!=0) {printf("KO\n Confirmacion
incorrecta\n");exit(1);}
            mensaje("OK\n");

              tiempo=(fin.tv_sec - inicio.tv_sec)*1000+(fin.tv_usec-inicio.tv_usec)/1000;
```

```
                mensaje(" Tiempo retardo (tx+rx) en ms = ");
                printf("%i\n",tiempo);
              }
            } else { // Paquete UDP perdido, o supera los 5 seg.
              printf("inf\n");
            }

          close(s_udp);
        }


        // El servidor necesita la dirección IP del cliente para mandar la confirmación. Si fuese
TCP la conseguiría
        // por su cuenta, pero en UDP se la debemos proporcionar.
        void servidor(int puertoUDP)
        {
          int s_udp, estado;
          char buf[100];
          struct sockaddr_in miaddr;
          struct sockaddr_in cliaddr;
          socklen_t temp;

          memset ( &miaddr, 0, sizeof (miaddr));
          miaddr.sin_family = AF_INET;
          miaddr.sin_port = htons(puertoUDP);
          miaddr.sin_addr.s_addr = INADDR_ANY;

          mensaje(" Servidor PINGv6 esperando conexión UDP ...\n");
          if ((s_udp=socket(AF_INET, SOCK_DGRAM, UDP))==-1) {printf("Error en
socket\n");exit(1);}
          if (bind(s_udp, (struct sockaddr *) &miaddr, sizeof(miaddr))==-1) {printf("error en
bind\n");exit(1);}


          while(1) {
            mensaje(" Conexión establecida. Esperando mensaje ... \n");

            estado=recvfrom(s_udp, buf, sizeof(buf), 0, &cliaddr, &temp);
            if (estado==-1) {printf("Error en recvfrom\n");exit(1);}
            mensaje(" Mensaje recibido. Comprobando FIRMA ... ");

            if (strncmp(buf, "PINGv6(CARLOS)", 14)!=0) {
              printf("KO\n Mensaje recibido incorrecto. Abortando ...\n");
              exit(1);
            }
            mensaje("OK\n");

            sprintf(buf, "CONFIRMADO\n");
            mensaje(" Mandando mensaje de confirmación\n");

            cliaddr.sin_family=AF_INET;
            cliaddr.sin_port=htons(puertoUDP);
            estado=sendto(s_udp, buf, sizeof (buf), 0, (struct sockaddr *) &cliaddr,
sizeof(cliaddr));
            if (estado==-1) {printf("Error en sendto\n");exit(1);}
          }
          close(s_udp);

        }
```